Manual Técnico

Strategos

30/12/2016

# Tabla de Contenido

[Tabla de Contenido 2](#_Toc470683089)

[CAPITULO I 3](#_Toc470683090)

[Modelo de Base de Datos 3](#_Toc470683091)

[CAPITULO II 40](#_Toc470683092)

[Arquitectura de Aplicación 40](#_Toc470683093)

[Estructura del Strategos Advance 40](#_Toc470683094)

[Requerimientos de Hardware y Software Comercial 41](#_Toc470683095)

[Configuración del Internet Explorer 43](#_Toc470683096)

[Configuración para Google Chrome 49](#_Toc470683097)

[Arquitectura General del Strategos Advance 54](#_Toc470683098)

[CAPITULO III 25](#_Toc470683099)

[INSTALACIÓN Y CONFIGURACIÓN DEL AMBIENTE DE DESARROLLO EN PLATAFORMA WINDOWS 25](#_Toc470683100)

[Instalación del Ambiente de Desarrollo 26](#_Toc470683101)

[Plataforma con que fue Desarrollado el Strategos Advance 31](#_Toc470683102)

[Framework Empleado para Construir el Strategos 32](#_Toc470683103)

[Cargar la Aplicación en el Ambiente de Desarrollo 33](#_Toc470683104)

[Capítulo V Estándares Visuales 51](#_Toc470683105)

[Convenciones Utilizadas 51](#_Toc470683106)

[Componentes 52](#_Toc470683107)

[Contenedores 52](#_Toc470683108)

[Visores 53](#_Toc470683109)

[Exploradores 54](#_Toc470683110)

[Buscadores 56](#_Toc470683111)

[Detalle de los Componentes Visuales del Strategos Advance 58](#_Toc470683112)

[Anexo I Diagrama Entidad Relación Base de Datos 85](#_Toc470683113)

#### Propósito del Manual

El siguiente documento tiene como propósito fundamental, la capacitación de conocimientos técnicos en cuanto a la estructura de la aplicación Strategos Advance, al personal de Sistemas de la organización. En este manual se presenta información relativa a las características técnicas del sistema, los requerimientos de hardware y software comercial, la arquitectura de la aplicación y de la base de datos, entre otros elementos técnicos.

# CAPITULO I

## Modelo de Base de Datos

#### Creación de la Base de Datos del Strategos Advance

1. Crear una Base de Datos, preferiblemente llamada Strategos. Turbinas soportadas: a) Oracle 9.x hasta 12.c, b) SQL Server 2008 hasta 2012, c) Postgres SQL 8.4 hasta 9.3.

2. Ejecutar los Script de creación de la Base de Datos en la siguiente secuencia:

1. Master SQL

2. Data SQL

#### Descripción de las Tablas del Sistema

| **Database Name** |
| --- |
| Strategos |

Listado de las tablas (entidades) que conforman la base de datos del Strategos, ordenadas alfabéticamente:

| **Entidades (Tablas)** | |
| --- | --- |
| **Nombre** | **Descripción** |
| accion | Tabla de Acciones Correctivas módulo de problemas |
| afw\_auditoria\_entero |  |
| afw\_auditoria\_evento |  |
| afw\_auditoria\_fecha | Tablas internas para registro de las acciones hechas por los usuarios, desplegables en el módulo de Auditoría |
| afw\_auditoria\_flotante |
| afw\_auditoria\_memo |
| afw\_auditoria\_string |
| afw\_objeto\_auditable |
| afw\_objeto\_auditable\_atributo |
| afw\_config\_usuario | Almacena para cada usuario la configuración de los paneles, tamaño de los paneles de indicadores, iniciativas, portafolios y panel de control. |
| afw\_configuracion | Todas las configuraciones Generales de Strategos, como la configuración de los colores de la pantalla, inserción de logos del cliente en la página principal del sistema, la configuración del servicio de cálculo de indicadores, la configuración de servidor de correo (email), tipo de autentificación de usuarios (LDAP), configuración del Login, se graban en esta tabla. |
| afw\_error | Registro de Errores del Sistema. Cada vez que se presenta un error, se guarda la información, a fin de que el fabricante del Strategos pueda darle solución. |
| afw\_grupo | Tabla Interna módulo de Seguridad. Almacena los nombres de los Grupos de Usuarios (roles). |
| afw\_permiso\_grupo | Tabla interna manejo de la seguridad. Permisología. Almacena cuáles permisos están asociados a un grupo (rol) de usuarios. |
| afw\_permiso | Tabla interna manejo de la seguridad. Permisología. Almacena cuáles funcionalidades del sistema está habilitadas para un determinado Rol (grupo de usuarios). |
| afw\_importacion | Tabla que almacena la configuración de las opciones de importación de data al Strategos desde fuentes externas. |
| afw\_modulo | Tabla que almacena los módulos que están inactivos en el Strategos. Dependiendo del cliente se pueden activar o desactivar funcionalidades del sistema. |
| afw\_pwd\_historia | Almacena todas las claves (password) asociadas a los usuarios, histórico de claves utilizadas por un usuario. |
| afw\_sistema | Almacena datos de la versión que está instalada, turbina de base de datos con la que se está conectando y licencia encriptada. |
| afw\_servicio | Almacena los resultados o log de los cálculos realizados por los usuarios. Parámetros del cálculo, fecha y hora de ejecución, períodos calculados y el nombre de indicadores que no pudo calcular. |
| afw\_user\_session | Registro de Sesión del usuario. Fecha y hora de conexión, usuario conectado, dirección IP. |
| afw\_usuario | Usuarios registrados en la base de datos con permisología para acceder al sistema. |
| afw\_usuario\_grupo | Roles. Almacena los grupos o roles, perfiles de acceso a los distintos módulos del sistema asociados a cada usuario. |
| afw\_version | Almacena el número de versión y el número built que está instalado actualmente. |
| categoria | Categorías de Medición utilizada para los indicadores de tipo cualitativo. |
| categoria\_por\_indicador | Almacena las categorías o expresiones utilizadas en la definición de indicadores cualitativos |
| causa | Causas de Problemas (modulo acciones correctivas) |
| causa\_por\_problema | Causas asociadas a un problema en particular |
| celda | Celdas de una Presentación Ejecutiva (panel de control) |
| clase | Clases de Indicadores |
| clase\_problema | Grupos o clases de problemas |
| conjunto\_formula | Almacena los ID y el tipo de serie (real, programado) de los indicadores que conforman a una fórmula de un indicador tipo FORMULA. |
| cuenta | Cuentas de Presupuesto |
| estado\_acciones | Tabla de estatus de acciones correctivas |
| duppont | Almacena para cada usuario a indicador tipo formula, la configuración de los árboles de Dupont (número de niveles de profundidad y año-período por defecto). |
| estado\_acciones | Almacena los posibles estados en que se puede encontrar una acción correctiva vinculada a un problema. |
| explicacion | Almacena las explicaciones (comentarios) que hacen los responsables de los indicadores e iniciativas (proyectos). |
| explicacion\_adjunto | Anexos (URL) de los anexos a las explicaciones de un objeto (indicador, iniciativa) |
| explicacion\_memo | Almacena los campos tipo memo de una explicación vinculada a los indicadores o iniciativas. (Descripción de la desviación, causas de retrasos, estrategias de solución). |
| formula | Almacena la expresión que define a la fórmula de los indicadores de tipo FORMULA. |
| grafico | Almacena la configuración que hace un usuario sobre el gráfico de un indicador. (Colores de las series, año y periodo de graficación, nombre del gráfico, etc.) |
| inc\_actividad | Almacena las Actividades por Iniciativa (proyecto). |
| indicador | Tabla de datos básicos de los indicadores. |
| indicador\_estado | Almacena los porcentajes de logro parcial y anual (estado) de los indicadores asociados a un plan. |
| indicador\_por\_iniciativa | Almacena los ID de los indicadores que están asociados a una iniciativa (Avance, Presupuesto, Eficiencia, Eficacia y otros definidos por los usuarios) |
| indicador\_por\_perspectiva | Almacena los ID de los indicadores asociados a una perspectiva (dimensión) de un plan |
| indicador\_por\_plan | Almacena los ID de los indicadores asociados a una perspectiva (dimensión) de un plan, así como el peso (importancia) del indicador. |
| indicador\_por\_portafolio | Almacena los ID de los indicadores asociados a un Portafolio de Iniciativas. (Avance del portafolio) |
| iniciativa | Almacena los Datos Básicos de las Iniciativas. |
| iniciativa\_ano | Almacena el porcentaje de avance al cierre de cada año para una iniciativa. |
| iniciativa\_estatus | Tabla de Estatus de las Iniciativas (posibles estados en los que se puede encontrar una iniciativa). |
| iniciativa\_plan | Almacena los ID de las iniciativas que están asociadas a un Plan. |
| iniciativa\_por\_perspectiva | Almacena los ID de las iniciativas vinculadas a una perspectiva (dimensión) de un plan. |
| masc\_cod\_plan\_cuentas\_grupo | Almacena la Mascará o formato del grupo de las cuentas de presupuesto |
| mascara\_codigo\_plan\_cuentas | Mascará o formato de las cuentas de presupuesto |
| medicion | Almacena las Mediciones de los Indicadores para cada año y período. |
| meta | Registro de las Metas de los Indicadores asociados a un Plan |
| metodologia | Tabla que almacena las metodologías de planificación (Balanced Scorecard, Marco Lógico, métodos de planificación propios de la empresa). |
| metodologia\_template | Almacena para cada metodología de planificación los nombres y orden de los niveles que la conforman. (Perspectivas-Objetivos, etc.) |
| modelo | Almacena los datos básicos de los modelos o mapas causa efecto creados por los usuarios. |
| objetos\_calcular | Almacena los Store Procedures del cálculo de indicadores e iniciativas. |
| organizacion | Almacena los Datos básicos de las organizaciones (punto focal de entrada al sistema). |
| organizacion\_memo | Campos de descripción de las organizaciones (descripción de la organización, visión, misión, etc.) |
| pagina | Almacena los datos básicos de las Páginas que conforman a de un Panel de Control (presentación ejecutiva). Nombre de la página, ancho y alto de las celdas, etc. |
| perspectiva | Almacena las Perspectivas (dimensiones estratégicas) de un plan. |
| perspectiva\_nivel | Almacena los porcentajes de logro de las perspectivas (dimensiones) de un plan para cada año y período de vigencia del plan. |
| perspectiva\_relacion | Almacena el orden de las perspectivas (dimensiones) de un plan. |
| plan\_nivel | Almacena los porcentajes de logro consolidado de un plan para cada año y período de vigencia del plan. |
| planes | Almacena los datos básicos de los planes estratégicos y operativos. |
| portafolio | Almacena los datos básicos de los portafolios de proyectos. |
| portafolio\_iniciativa | Almacena los ID y el peso (importancia) de las iniciativas que están asociadas a un portafolio. |
| problema | Almacena los datos básicos de los problemas. Módulo de Acciones Correctivas. |
| problema\_memo | Almacena los campos de descripción de un Problema. Módulo de Acciones Correctivas. |
| pry\_actividad | Almacena las actividades de una iniciativa. |
| pry\_calendario | Almacena los días laborables de una iniciativa. |
| pry\_calendario\_detalle | Almacena los días no laborables empleados en el calendario de las iniciativas. |
| reporte | Almacena los datos básicos de los reportes creados por los usuarios en el módulo de VISTAS DE DATOS. |
| responsable | Almacena los Responsables que serán asociados a los indicadores, iniciativas y sus actividades. |
| responsable\_por\_accion | Almacena los ID de los responsables de las acciones correctivas vinculadas a un problema. |
| seguimiento | Almacena los reportes de seguimiento a las acciones correctivas asociadas a un problema. |
| seguimiento\_memo | Almacena los campos memos de un reporte de seguimiento asociado a una acción correctiva de un problema. |
| seguimiento\_mensaje\_email | Almacena la configuración del mensaje de correo electrónico que será enviado a los responsables de ejecutar una acción correctiva vinculada a un problema. |
| serie\_indicador | Almacena las series de tiempo vinculadas a los indicadores. |
| serie\_tiempo | Almacena la tabla general de series de tiempo (real, programado, mínimo y máximo) |
| unidad | Almacena la Tabla de Unidades de Medida que serán utilizadas para medir a los indicadores. |
| vista | Almacena los datos básicos de los Paneles de control (Presentaciones Ejecutivas). |

**Atributos por Entidad**

**Módulo de Seguridad y Parametrización (tablas básicas):**

|  |  |
| --- | --- |
| afw\_auditoria\_entero |  |
| afw\_auditoria\_evento |  |
| afw\_auditoria\_fecha | Tablas internas para registro de las acciones hechas por los usuarios, desplegables en el módulo de Auditoría |
| afw\_auditoria\_flotante |
| afw\_auditoria\_memo |
| afw\_auditoria\_string |
| afw\_objeto\_auditable |
| afw\_objeto\_auditable\_atributo |

| Attribute(s) of "afw\_auditoria\_entero" Acciones realizadas por los usuarios sobre los campos de la base de datos de tipo ENTERO | | |
| --- | --- | --- |
| **Atributo** | **Tipo de Dato** | **Descripción** |
| fecha | datetime | Fecha de la instancia |
| instancia\_id | varchar(100) | Identificador único de la instancia |
| nombre\_atributo | varchar(100) | Nombre del campo o atributo modificado o creado |
| valor | numeric(10,0) | Valor modificado o creado campo tipo entero |
| usuario\_id | numeric(10,0) | Identificador único del usuario que ejecutó la acción |
| objeto\_id | numeric(10,0) | Identificador único del objeto modificado o creado |
| tipo\_evento | numeric(1,0) | Tipo de acción ejecutada |
| valor\_anterior | numeric(10,0) | Valor que tenía el registro antes de la modificación |

| Attribute(s) of "afw\_auditoria\_evento" | | |
| --- | --- | --- |
| **Name** | **Datatype** | **Definition** |
| fecha | datetime | Fecha de la instancia |
| instancia\_id | varchar(100) | Identificador único de la instancia |
| tipo\_evento | numeric(1,0) | Tipo de acción ejecutada |
| objeto\_id | numeric(10,0) | Identificador único del objeto modificado o creado |
| instancia\_nombre | varchar(200) | Descripción de la acción ejecutada por el usuario |
| usuario\_id | numeric(10,0) | Identificador único del usuario que ejecutó la acción sobre el objeto auditado |

| Attribute(s) of "afw\_auditoria\_fecha" " Acciones realizadas por los usuarios sobre los campos de la base de datos de tipo FECHA datetime | | |
| --- | --- | --- |
| **Name** | **Datatype** | **Definition** |
| fecha | datetime | Fecha de la instancia |
| instancia\_id | varchar(100) | Identificador único de la instancia |
| nombre\_atributo | varchar(100) | Nombre del campo modificado o creado |
| valor | datetime | Valor modificado o creado del campo tipo fecha |
| usuario\_id | numeric(10,0) | Identificador único del usuario |
| objeto\_id | numeric(10,0) | Identificador único del objeto modificado o creado |
| tipo\_evento | numeric(1,0) | Tipo de acción ejecutada |
| valor\_anterior | datetime | Valor anterior a la modificación |

| Attribute(s) of "afw\_auditoria\_flotante" " Acciones realizadas por los usuarios sobre los campos de la base de datos de tipo FLOAT | | |
| --- | --- | --- |
| **Name** | **Datatype** | **Definition** |
| fecha | datetime | Fecha de la instancia |
| instancia\_id | varchar(100) | Identificador único de la instancia |
| nombre\_atributo | varchar(100) | Nombre del campo modificado o creado |
| valor | float | Valor modificado o creado |
| usuario\_id | numeric(10,0) | Identificador único del usuario |
| objeto\_id | numeric(10,0) | Identificador único del objeto modificado o creado |
| tipo\_evento | numeric(1,0) | Tipo de acción ejecutada |
| valor\_anterior | float | Valor anterior a la modificación |

| Attribute(s) of "afw\_auditoria\_memo" " Acciones realizadas por los usuarios sobre los campos de la base de datos de tipo VARCHAR | | |
| --- | --- | --- |
| **Name** | **Datatype** | **Definition** |
| fecha | datetime | Fecha de la instancia |
| instancia\_id | varchar(100) | Identificador único de la instancia |
| nombre\_atributo | varchar(100) | Nombre del campo modificado o creado |
| valor | text | Contenido modificado o creado campo tipo texto |
| usuario\_id | numeric(10,0) | Identificador único del usuario |
| objeto\_id | numeric(10,0) | Identificador único del objeto modificado o creado |
| tipo\_evento | numeric(1,0) | Tipo de acción ejecutada |
| valor\_anterior | text | Contenido anterior a la modificación |

| Attribute(s) of "afw\_auditoria\_string" " Acciones realizadas por los usuarios sobre los campos de la base de datos de tipo TEXTO | | |
| --- | --- | --- |
| **Name** | **Datatype** | **Definition** |
| fecha | datetime | Fecha de la instancia |
| instancia\_id | varchar(100) | Identificador único de la instancia |
| nombre\_atributo | varchar(100) | Nombre del campo tipo string modificado o creado |
| valor | varchar(500) | Contenido del string modificado o creado |
| usuario\_id | numeric(10,0) | Identificador único del usuario |
| objeto\_id | numeric(10,0) | Identificador único del objeto modificado o creado |
| tipo\_evento | numeric(1,0) | Tipo de acción ejecutada |
| valor\_anterior | varchar(500) | Contenido anterior a la modificación |

|  |  |  |  |
| --- | --- | --- | --- |
| afw\_config\_usuario | | Almacena para cada usuario la configuración de los paneles, tamaño de los paneles de indicadores, iniciativas, portafolios y panel de control. | |
| Attribute(s) of "afw\_config\_usuario" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| usuario\_id | | numeric(10,0) | | Identificador único del usuario |
| objeto | | varchar(100) | | Descripción del objeto (indicador, proyecto, etc.) |
| configuracion\_base | | varchar(200) | | Formato de configuración del objeto |
| data | | text | | Tamaño del panel en el visor del objeto |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| afw\_configuracion | | | Almacena todas las configuraciones Generales de Strategos, como la configuración de los colores de la pantalla, inserción de logos del cliente en la página principal del sistema, la configuración del servicio de cálculo de indicadores, la configuración de servidor de correo (email), tipo de autentificación de usuarios (LDAP), configuración del Login, se graban en esta tabla. | |
| Attribute(s) of "afw\_configuracion" | | | | |
| **Name** | **Datatype** | | **Definition** | |
| parametro | varchar(200) | | Nombre del Parámetro de la configuración de la sesión de usuario. | |
| valor | text | | Valor del parámetro. Configuración del sistema, color de fondo, conexión por LDAP o propia, correo del sistema, configuración del servicio (base de datos), columnas de los visores, etc.  Configuración Global  Alertas de los planes, si muestra los indicadores de eficiencia, eficacia, nombres de los indicadores de los proyectos. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| afw\_error | | Registro de Errores del Sistema. Cada vez que se presenta un error, se guarda la información, a fin de que el fabricante del Strategos pueda darle solución. | | |
| Attribute(s) of "afw\_error" Entity (Tabla Interna Registro de Errores del Sistema) | | | | |
| **Name** | | **Datatype** | | **Definition** |
| err\_number | | numeric(10,0) | | Nro. del Error ocurrido |
| err\_description | | varchar(2000) | | Descripción del error |
| err\_timestamp | | datetime | | Fecha de ocurrencia |
| err\_user\_id | | varchar(20) | | Identificador único del usuario que estaba conectado. |
| err\_version | | varchar(20) | | Descripción de la Versión del sistema |
| err\_stacktrace | | text | | Descripción de la traza que dejo el error |
| err\_cause | | varchar(2000) | | Descripción de la Causa del error |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| afw\_grupo | | Tabla Interna módulo de Seguridad. Almacena los nombres de los Grupos de Usuarios (roles). | | |
| Attribute(s) of "afw\_grupo" Entity | | | | |
| **Name** | | **Datatype** | | **Definition** |
| grupo\_id | | numeric(10,0) | | Identificador único del grupo (rol) |
| grupo | | varchar(50) | | nombre del grupo (rol) |
| creado | | datetime | | Fecha de creación |
| modificado | | datetime | | Fecha de modificación |
| creado\_id | | numeric(10,0) | | Identificador único del usuario que creó el registro |
| modificado\_id | | numeric(10,0) | | Identificador único del usuario que modificó el registro |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| afw\_importacion | | Tabla que almacena la configuración de las opciones de importación de data al Strategos desde fuentes externas. | | |
| Attribute(s) of "afw\_importacion" Entity | | | | |
| **Name** | | **Datatype** | | **Definition** |
| id | | number(10) | | Id del registro |
| nombre | | varchar2(100) | | Nombre del proceso de importación |
| usuario\_id | | number(10) | | ID del usuario asociado al registro |
| tipo | | number(1) | | Tipo de Importación |
| configuracion | | varchar(2000) | | Nombre de la fuente de datos para importar y su ubicación, tipo de serie de tiempo a importar, año y período a importar |

|  |  |  |  |
| --- | --- | --- | --- |
| afw\_modulo | | Tabla que almacena los módulos que están inactivos en el Strategos. Dependiendo del cliente se pueden activar o desactivar funcionalidades del sistema. | |
| Attribute(s) of "afw\_modulo" Entity | | | | |
| **Name** | | **Datatype** | | **Definition** |
| Id | | varchar(50) | | ID del registro |
| Modulo | | varchar (100) | | Nombre del Módulo (Indicadores, Planes, etc.) |
| Activo | | number(1) | | Flag que indica si el módulo está activo o no |
| Nombre | | varchar2(50) | | Descripción del Módulo |

|  |  |  |  |
| --- | --- | --- | --- |
| afw\_objeto\_auditable | | Tabla que almacena los objetos del sistema que son susceptibles a registrarle movimientos para auditoría | |
| Attribute(s) of "afw\_objeto\_auditable " | | | | |
| **Name** | | **Datatype** | | **Definition** |
| objeto\_id | | number(10) | | Identificador del Objeto (indicador, iniciativa, plan, etc.) |
| nombre\_clase | | varchar2(200) | | Nombre de la clase Java a la que pertenece el objeto |
| nombre\_campo\_id | | varchar2 (100) | | Nombre del campo a ser auditado |
| nombre\_campo\_nombre | | varchar2 (100) | | Etiqueta para el nombre del campo auditado |
| auditoria\_activa | | number (1) | | Flag que indica si la auditoría está o no activa para este objeto y campo. |

| Attribute(s) of "afw\_objeto\_auditable\_atributo" Entity | | |
| --- | --- | --- |
| **Name** | **Datatype** | **Definition** |
| objeto\_id | numeric(10,0) | Identificador único del objeto |
| nombre\_atributo | varchar(50) | Nombre del atributo que será auditado |
| tipo | numeric(1,0) | Tipo de Acción (ingreso, modificación, eliminación, etc.) |
| configuracion | varchar(2000) | Descrición del atributo |

|  |  |  |  |
| --- | --- | --- | --- |
| afw\_permiso | | Tabla interna manejo de la seguridad. Permisología. Almacena cuáles funcionalidades del sistema está habilitadas para un determinado Rol (grupo de usuarios). | |
| Attribute(s) of "afw\_permiso" Entity | | | | |
| **Name** | | **Datatype** | | **Definition** |
| permiso\_id | | varchar(50) | | Identificador único del permiso |
| padre\_id | | varchar(50) | | Identificador único del permiso padre (jerarquía) |
| permiso | | varchar(50) | | Nombre del permiso (módulo del sistema) |
| nivel | | numeric(2,0) | | Nivel dentro del árbol de permisos |
| grupo | | numeric(2,0) | | Grupo al que pertenece |
| global | | numeric(1,0) | | Permiso global o local |
| descripcion | | varchar(2000) | | Descripción del permiso |

|  |  |  |  |
| --- | --- | --- | --- |
| afw\_permiso\_grupo | | Tabla interna manejo de la seguridad. Permisología. Almacena cuáles permisos están asociados a un grupo (rol) de usuarios. | |
| Attribute(s) of "afw\_permiso\_grupo" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| permiso\_id | | varchar(50) | | Identificador único del permiso |
| grupo\_id | | numeric(10,0) | | Identificador único del grupo (rol) al que pertenece el permiso |

|  |  |  |  |
| --- | --- | --- | --- |
| afw\_pwd\_historia | | Almacena todas las claves (password) asociadas a los usuarios, histórico de claves utilizadas por un usuario. | |
| Attribute(s) of "afw\_pwd\_historia" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| usuario\_id | | number(10) | | ID del Usuario |
| fecha | | timestamp | | Fecha de asignación de la clave (password) |
| pwd | | varchar(100) | | Password creado por el usuario (encriptado) |

|  |  |  |  |
| --- | --- | --- | --- |
| afw\_servicio | | Almacena los resultados o log de los cálculos realizados por los usuarios. Parámetros del cálculo, fecha y hora de ejecución, períodos calculados y el nombre de indicadores que no pudo calcular. | |
| Attribute(s) of "afw\_servicio" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| usuario\_id | | number(10) | | ID del usuario que ejecutó el proceso de calcular |
| fecha | | timestamp | | Fecha de ejecución del proceso de cálculo |
| nombre | | varchar (50) | | Nombre del proceso |
| estatus | | number (1) | | Estado del proceso (ejecutado, revisado, etc.) |
| mensaje | | varchar 2(1000) | | Mensaje generado por el sistema con los resultados del cálculo |
| log | | Long varchar | | Descripción de los indicadores no calculados |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| afw\_sistema | | Almacena datos de la versión que está instalada, turbina de base de datos con la que se está conectando y licencia encriptada. | | |
| Attribute(s) of "afw\_sistema" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| id | | numeric(10,0) | | Identificador único del registro |
| version | | varchar(20) | | Numero de versión del Strategos instalada |
| creado | | datetime | | Fecha de creación del registro |
| modificado | | datetime | | Fecha de modificación del registro |
| rdbms\_id | | varchar(20) | | Tipo de Base de datos (PostGrest, Sql Server, Oracle, etc.) |
| producto | | varchar(50) | | Producto instalado (Strategos, Radar, …) |
| cmaxc | | Varchar2(1000) | | Serial de instalación (Licencia) asignada al cliente (encriptada) |
| conexion | | Varchar2(50) | | Parámetros de conexión a la base de datos |
| build | | numeric(10,0) | | Numero de construcción del producto (versión) |
| actual | |  | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| afw\_user\_session | | Registro de Sesión del usuario. Fecha y hora de conexión, usuario conectado, dirección IP. | |
| Attribute(s) of "afw\_user\_session" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| session\_id | | varchar(100) | | Identificador único de la sesión |
| usuario\_id | | numeric(10,0) | | Identificador único del usuario que se conectó al sistema |
| login\_ts | | datetime | | Fecha del login |
| remote\_address | | varchar(200) | | Dirección IP |
| remote\_host | | varchar(200) | | Dirección del servidor |
| remote\_user | | varchar(1000) | | Descripción del usuario |
| url | | varchar(100) | | Dirección URL a la que se conectó |
| persona\_id | | number(10) | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| afw\_usuario | | Usuarios registrados en la base de datos con permisología para acceder al sistema. | |
| Attribute(s) of "afw\_usuario" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| usuario\_id | | numeric(10,0) | | Identificador único del usuario |
| full\_name | | varchar(50) | | Nombre del usuario |
| u\_id | | varchar(20) | | User ID |
| pwd | | varchar(100) | | Clave |
| is\_admin | | numeric(1,0) | | Flag que indica si el usuario es administrador |
| is\_connected | | numeric(1,0) | | Flag que indica si el usuario está conectado a la base de datos |
| time\_stamp | | datetime | | Fecha de conexión del usuario al sistema |
| creado | | datetime | | Fecha de creación del registro |
| modificado | | datetime | | Fecha de modificación del registro |
| creado\_id | | numeric(10,0) | | Identificador del usuario que creó el registro |
| modificado\_id | | numeric(10,0) | | Identificador del usuario que modificó el registro |
| instancia | | varchar(40) | | Descripción de la instancia |
| inactivo | | numeric(1,0) | | Flag que indica si el usuario está inactivo |
| estatus | | numeric(1,0) | | Estatus del usuario |
| bloqueado | | numeric(1,0) | | Flag que indica si el usuario está bloqueado |
| ultima\_modif\_pwd | | datetime | | Fecha de la última modificación del password |
| deshabilitado | | numeric(1,0) | | Flag de usuario deshabilitado para usar el sistema |
| forzarCambiarPwd | | numeric(1,0) | | Flag que indica que el usuario debe cambiar la clave a su primer login en el sistema |

|  |  |  |  |
| --- | --- | --- | --- |
| afw\_usuario\_grupo | | Roles. Almacena los grupos o roles, perfiles de acceso a los distintos módulos del sistema asociados a cada usuario. | |
| Attribute(s) of "afw\_usuario\_grupo" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| usuario\_id | | numeric(10,0) | | Identificador único del usuario |
| grupo\_id | | numeric(10,0) | | Identificador único del grupo o rol asociado al usuario |
| organizacion\_id | | numeric(10,0) | | Identificador único de la organización a la que pertenece el usuario con el grupo asociado |
| modificado | | datetime | | Fecha de modificación del registro del usuario |
| modificado\_id | | numeric(10,0) | | Identificador del usuario que modifico el registro |

|  |  |  |  |
| --- | --- | --- | --- |
| afw\_version | | Almacena el número de versión y el número built que está instalado actualmente. | |
| Attribute(s) of "afw\_version" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| version | | Varchar2(50) | | Número de la versión del Strategos que fue instalada |
| build | | Varchar2 (8) | | Número del Built |
| dateBuild | | Varchar2 (50) | | Fecha del Built |
| createdAt | | timestamp | | Fecha de instalación de la versión |

|  |  |  |  |
| --- | --- | --- | --- |
| categoria | | Categorías de Medición utilizada para los indicadores de tipo cualitativo. | |
| Attribute(s) of "categoria" Entity | | | | |
| **Name** | | **Datatype** | | **Definition** |
| categoria\_id | | numeric(10,0) | | Identificador único de la Categoría |
| nombre | | varchar(50) | | Descripción de la categoría de medición para el indicador de tipo cualitativo |

|  |  |  |  |
| --- | --- | --- | --- |
| causa | | Causas de Problemas (modulo acciones correctivas) | |
| Attribute(s) of "causa" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| causa\_id | | numeric(10,0) | | Identificador único de la Causa |
| padre\_id | | numeric(10,0) | | Identificador único de la causa padre (árbol) |
| nombre | | varchar(50) | | Nombre de la causa de retraso |
| descripcion | | text | | Descripción de la causa de retraso |
| nivel | | numeric(2,0) | | Nivel que ocupa la causa en el árbol de causas |

|  |  |  |  |
| --- | --- | --- | --- |
| cuenta | | Cuentas de Presupuesto | |
| Attribute(s) of "cuenta" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| cuenta\_id | | numeric(10,0) | | Identificador único de la cuenta de presupuesto |
| padre\_id | | numeric(10,0) | | Identificador único de la cuenta padre (árbol) |
| codigo | | varchar(10) | | Código que identifica a la cuenta |
| descripcion | | varchar(250) | | Descripción de la cuenta |

|  |  |  |  |
| --- | --- | --- | --- |
| estado\_acciones | | Tabla de estados posibles de las acciones correctivas (módulo de problemas) | |
| Attribute(s) of "estado\_acciones" | | |
| **Name** | **Datatype** | **Definition** |
| estado\_id | numeric(10,0) | Identificador único del estatus de la acción |
| nombre | varchar(15) | Nombre del estado |
| aplica\_seguimiento | numeric(1,0) | Flag que indica si se debe hacer seguimiento a la acción correctiva cuando se encuentre en este estado |
| orden | numeric(4,0) | Orden del estatus en la lista de estatus creados |
| condicion | numeric(4,0) | Estatus de la acción correctiva |

|  |  |  |  |
| --- | --- | --- | --- |
| masc\_cod\_plan\_cuentas\_grupo | | Tabla de los formatos posibles de cuentas de presupuesto | |
| Attribute(s) of "masc\_cod\_plan\_cuentas\_grupo" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| nivel | | numeric(4,0) | | Número del grupo o nivel de la máscara |
| mascara\_id | | numeric(10,0) | | Identificación único de la máscara |
| mascara | | varchar(10) | | Código de la máscara para el nivel |
| Nombre | | varchar(50) | | Descripción de la máscara |

|  |  |  |  |
| --- | --- | --- | --- |
| mascara\_codigo\_plan\_cuentas | | Tabla de la estructura de las cuentas de presupuesto | |
| Attribute(s) of "mascara\_codigo\_plan\_cuentas" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| mascara\_id | | numeric(10,0) | | Identificador único de la máscara que tendrán las cuentas de presupuesto |
| Niveles | | numeric(4,0) | | Número de niveles de la máscara |

|  |  |  |  |
| --- | --- | --- | --- |
| iniciativa\_estatus | | Tabla contentiva de los posibles estados en los que se puede encontrar una iniciativa (proyecto) | |
| Attribute(s) of "iniciativa\_estatus" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| id | | numeric(10,0) | | Identificador único ID del estatus de la iniciativa (tabla de tipos de estatus) |
| nombre | | varchar(50) | | Nombre del estatus |
| porcentaje\_inicial | | float | | Representa el porcentaje de inicio de avance de la iniciativa para indicar que se encuentra en ese estatus. |
| porcentaje\_final | | float | | Representa el porcentaje final de avance de la iniciativa para indicar que se encuentra en ese estatus. Este y el campo anterior representan un rango dentro del cual se debe ubicar el porcentaje de avance de la iniciativa para que le sea asignado el estatus definido en este registro |
| sistema | | numeric(1) | | Flag que indica que este estatus es creado por el sistema y no podrá ser modificado por el usuario. |
| bloquear\_medicion | | numeric(1) | | Flag que indica que el sistema bloqueara las mediciones de la iniciativa si esta se encuentra en este estatus. Eso impide que el usuario registre avance o programado a una iniciativa que se encuentre en este estatus. |
| bloquear\_indicadores | | numeric(1) | | Flag que indica que el sistema bloqueara las mediciones de todos los indicadores de la iniciativa si esta se encuentra en este estatus. Eso impide que el usuario modifique o elimine información de los indicadores de avance, eficiencia, eficacia y presupuesto de una iniciativa que se encuentre en este estatus. |

|  |  |  |  |
| --- | --- | --- | --- |
| metodologia | | Tabla contentiva de las metodologías de planificación (Balanced Scorecard, Marco Lógico, etc.) | |
| Attribute(s) of "metodologia" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| metodologia\_id | | numeric(10,0) | | Identificador único de la metodología de planes |
| Nombre | | varchar(50) | | Nombre de la Metodología |
| Descripción | | text | | Descripción de la Metodología |
| Creado | | datetime | | Fecha de creación del registro |
| Modificado | | datetime | | Fecha de creación del registro |
| creado\_id | | numeric(10,0) | | Identificador único del usuario que creó el registro |
| modificado\_id | | numeric(10,0) | | Identificador único del usuario que modificó el registro |
| indicador\_nombre | | varchar(50) | | Nombre de la sección de la metodología asociada a los indicadores |
| iniciativa\_nombre | | varchar(50) | | Nombre de la sección de la metodología asociada a las iniciativas / proyectos |
| actividad\_nombre | | varchar(50) | | Nombre de la sección de la metodología asociada a las actividades de las iniciativas |
| indicador\_nombre\_plural | | varchar(50) | | Etiqueta en plural asociada al área de indicadores |
| iniciativa\_nombre\_plural | | varchar(50) | | Etiqueta en plural asociada al área de iniciativas |
| actividad\_nombre\_plural | | varchar(50) | | Etiqueta en plural asociada al área de actividades de las iniciativas |
| indicador\_articulo | | varchar(50) | | Etiqueta del artículo (el, las, etc.) empleado para el área de indicadores |
| iniciativa\_articulo | | varchar(50) | | Etiqueta del artículo (el, las, etc.) empleado para el área de iniciativas |
| actividad\_articulo | | varchar(50) | | Etiqueta del artículo (el, las, etc.) empleado para el área de actividades |
| indicador\_articulo\_plural | | varchar(50) | | Etiqueta del artículo (el, las, etc.) en plural empleado para el área de indicadores |
| iniciativa\_articulo\_plural | | varchar(50) | | Etiqueta del artículo (el, las, etc.) en plural empleado para el área de iniciativas |
| actividad\_articulo\_plural | | varchar(50) | | Etiqueta del artículo (el, las, etc.) en plural empleado para el área de actividades |

|  |  |  |  |
| --- | --- | --- | --- |
| metodologia\_template | | Tabla contentiva de la estructura de una metodología de planificación (en el caso del Balanced Scorecard, la estructura sería Perspectivas-Objetivos-Indicador Estratégico-Iniciativas) | |
| Attribute(s) of "metodologia\_template" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| metodologia\_template\_id | | numeric(10,0) | | Identificador único de la planilla de planes |
| Nombre | | varchar(50) | | Nombre de la Metodología de Planificación |
| metodologia\_id | | numeric(10,0) | | Identificador único de la metodología de planificación |
| Orden | | numeric(4,0) | | Orden del elemento dentro de la plantilla |
| Tipo | | numeric(2,0) | | Tipo de dimensión (perspectiva, objetivo, etc.) |

|  |  |  |  |
| --- | --- | --- | --- |
| objetos\_calcular | | Tabla contentiva de los Procedimientos Almacenados (Store Procedures) para el cálculo de los indicadores e iniciativas | |
| Attribute(s) of "objetos\_calcular" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| Id | | numeric(10,0) | |  |
| Objeto\_Id | | numeric(10,0) | |  |
| Usuario\_Id | | numeric(10,0) | |  |
| Calculado | | numeric(1) | |  |
| organizacion | | Tabla contentiva de los datos básicos de las Organizaciones y la estructura de organizaciones de un grupo de empresas - empresa – institución, incluye departamentos, gerencias, unidades, etc. | |
| Attribute(s) of "organizacion" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| organizacion\_id | | numeric(10,0) | | Identificador único de la Organización |
| padre\_id | | numeric(10,0) | | Identificador único de la organización padre de la que se está creando (jerarquía) |
| nombre | | varchar(150) | | Nombre de la organización |
| rif | | varchar(15) | | Registro de Información Fiscal (solo Venezuela) |
| direccion | | varchar(150) | | Dirección de la sede de la organización |
| telefono | | varchar(50) | | Teléfonos de la organización |
| fax | | varchar(50) | | Nro. de Fax |
| mes\_cierre | | numeric(4,0) | | Mes en que cierra el año fiscal de la organización |
| creado | | datetime | | Fecha de creación del registro |
| modificado | | datetime | | Fecha de creación del registro |
| creado\_id | | numeric(10,0) | | Identificación única del usuario que creó el registro |
| modificado\_id | | numeric(10,0) | | Identificación única del usuario que modificó el registro |
| alerta\_min\_max | | numeric(4,0) | | % de Alerta que se aplicará a los indicadores de tipo mínimo y máximo que pertenezcan a la organización |
| alerta\_meta\_n1 | | numeric(4,0) | | % de Alerta que se aplicará a los indicadores de tipo meta Inferior que pertenezcan a la organización |
| alerta\_meta\_n2 | | numeric(4,0) | | % de Alerta que se aplicará a los indicadores de tipo meta superior que pertenezcan a la organización |
| alerta\_iniciativa\_zv | | numeric(4,0) | | % de Alerta que se aplicará a las iniciativas zona verde |
| alerta\_iniciativa\_za | | numeric(4,0) | | % de Alerta que se aplicará a las iniciativas zona amarilla |
| enlace\_parcial | | varchar(20) | | Descripción del código parcial de enlace utilizado para la importación de datos |
| visible | | numeric(1,0) | | Flag que indica si el registro es visible o no al usuario |
| read\_only | | numeric(1,0) | | Flag que indica si el registro es de solo lectura |

|  |  |  |  |
| --- | --- | --- | --- |
| organización\_memo | | Tabla contentiva de los datos tipo memo asociados a una organización, tales como descripción, misión, visión, lineamientos estratégicos, valores, etc. | |
| Attribute(s) of "organizacion\_memo" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| organizacion\_id | | numeric(10,0) | | Identificación único de la organización |
| tipo | | numeric(2,0) | | Tipo de Campo memo (Visión, Misión, Valores, Lineamientos Estratégicos, etc. |
| descripcion | | text | | Descripción del campo memo |

|  |  |  |  |
| --- | --- | --- | --- |
| responsable | | Tabla contentiva de los datos básicos de los responsables de ejecutar las iniciativas (proyectos), y de gestionar los indicadores. Responsables de lograr las metas, de hacer seguimiento, etc. | |
| Attribute(s) of "responsable" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| responsable\_id | | numeric(10,0) | | Identificador único del responsable |
| usuario\_id | | numeric(10,0) | | Identificador único de del usuario de Strategos asociado al responsable |
| nombre | | varchar(50) | | Nombre del responsable |
| cargo | | varchar(50) | | Cargo que ocupa en la organización |
| ubicacion | | varchar(250) | | Ubicación física del responsable en la Organización |
| email | | varchar(50) | | Correo electrónico del responsable |
| notas | | text | | Campo de descripción |
| children\_count | | numeric(1,0) | | Contador interno |
| Tipo | | numeric(1,0) | | Tipo de responsable (ordinario o grupo) |
| grupo | | numeric(1,0) | | Flag que indica si el responsable pertenece a un Grupo |
| organizacion\_id | | numeric(10,0) | | Identificador único de la organización a la que pertenece el responsable |

|  |  |  |  |
| --- | --- | --- | --- |
| series\_tiempo | | Tabla contentiva de las series de tiempo que serán utilizadas para controlar los indicadores (real ejecutado, programado, mínimo, máximo, meta) | |
| Attribute(s) of "serie\_tiempo" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| serie\_id | | numeric(10,0) | | Identificador único de la serie de tiempo |
| nombre | | varchar(50) | | Descripción de la serie de tiempo |
| fija | | numeric(1,0) | | Tipo de serie (fija, no fija) |
| oculta | | numeric(1,0) | | Flag si es visible o no al usuario |
| identificador | | varchar(5) | | Código de la serie de tiempo |
| preseleccionada | | numeric(1,0) | | Flag si es o no preseleccionada |

|  |  |  |  |
| --- | --- | --- | --- |
| unidad | | Tabla contentiva de las unidades que serán utilizadas para medir a los indicadores (%, numero, dólares, kilos, etc.) | |
| Attribute(s) of "unidad" | | | | |
| **Name** | | **Datatype** | | **Definition** |
| unidad\_id | | numeric(10,0) | | Identificador único de la Unidad |
| nombre | | varchar(100) | | Descripción de la unidad de medida |
| tipo | | numeric(2,0) | | Tipo de unidad |

**Módulo de Indicadores:**

|  |  |  |  |
| --- | --- | --- | --- |
| clase | | Almacena las Clases de Indicadores | |
| Attribute(s) of "clase" | | | |
| **Name** | **Datatype** | | **Definition** |
| clase\_id | numeric(10,0) | | Identificador único de la Clase de Indicador |
| padre\_id | numeric(10,0) | | Identificador único de la Clase Padre (árbol) de Indicador |
| organizacion\_id | numeric(10,0) | | Identificador único de la Organización a la que pertenece la Clase de Indicador |
| nombre | varchar(310) | | Nombre de la Clase |
| descripcion | varchar(250) | | Descripción de la Clase |
| creado | datetime | | Fecha de creación del registro |
| modificado | datetime | | Fecha de creación del registro |
| modificado\_id | numeric(10,0) | | Identificador único del usuario que modificó el registro |
| creado\_id | numeric(10,0) | | Identificador único del usuario que creó el registro |
| enlace\_parcial | varchar(20) | | Código parcial de enlace para el proceso de importación de datos |
| tipo | numeric(2,0) | | Tipo de Clase, indica si la clase está asociada al módulo de indicadores, planes o iniciativas (proyectos) |
| visible | numeric(1,0) | | Flag de clase visible o no al usuario. Hay clases internas que utiliza el sistema para almacenar información de ciertos indicadores (logro de los planes, avance de las iniciativas, etc.) |

|  |  |  |  |
| --- | --- | --- | --- |
| indicador | | Tabla de datos básicos de los indicadores | |
| Attribute(s) of "indicador" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| indicador\_id | numeric(10,0) | | Identificador único del indicador |
| organizacion\_id | numeric(10,0) | | Identificador único de la organización a la que pertenece el indicador |
| clase\_id | numeric(10,0) | | Identificador único de la clase a la que pertenece el indicador |
| unidad\_id | numeric(10,0) | | Identificador único de la unidad de medida asociada al indicador |
| nombre | varchar(150) | | Nombre del Indicador |
| descripcion | varchar(2000) | | Descripción del Indicador |
| naturaleza | numeric(1,0) | | Naturaleza (simple, formula, sumatoria, etc.) |
| frecuencia | numeric(1,0) | | Frecuencia de medición (diario, mensual, etc.) |
| guia | varchar(50) | | Flag que indique si el indicador de tipo Guía o Resultado (metodología del BSC) |
| codigo\_enlace | varchar(100) | | Código de enlace del indicador. Importación de datos |
| nombre\_corto | varchar(150) | | Nombre corto del indicador (árboles de Dupont) |
| prioridad | numeric(1,0) | | Prioridad del indicador |
| constante | numeric(1,0) | | Flag que indica si el indicador es una constante |
| fuente | char varying(2000) | | Descripción del origen del dato que alimenta al indicador |
| orden | varchar(10) | | Orden del indicador dentro de la clase |
| read\_only | numeric(1,0) | | Flag de solo lectura, el indicador no podrá ser modificado |
| caracteristica | numeric(1,0) | | Descripción de la característica del indicador. Reto de aumento, disminución, condición de valor máximo o mínimo |
| crecimiento\_ind | numeric(1,0) | | Alerta del Indicador en el Plan ultima alerta |
| fecha\_ultima\_medicion | varchar(10) | | Fecha de la última medición registrada para el indicador |
| ultima\_medicion | float | | Valor de la última medición registrada para el indicador |
| resp\_fijar\_meta\_id | numeric(10,0) | | Identificador único del responsable de fijar la meta del indicador |
| resp\_lograr\_meta\_id | numeric(10,0) | | Identificador único del responsable de lograr la meta del indicador |
| resp\_seguimiento\_id | numeric(10,0) | | Identificador único del responsable de hacer seguimiento al indicador |
| resp\_cargar\_meta\_id | numeric(10,0) | | Identificador único del responsable de cargar la meta del indicador |
| resp\_cargar\_ejecutado\_id | numeric(10,0) | | Identificador único del responsable de cargar la data ejecutada del indicador |
| lag\_lead | numeric(1,0) | | Flag que indica si el indicador es tipo guía o resultado metodología del BSC. |
| corte | numeric(1,0) | | Tipo de corte longitudinal o transversal |
| enlace\_parcial | varchar(20) | | Descripción del código parcial de enlace para la importación de datos al indicador |
| alerta\_n2\_ind\_id | numeric(10,0) | | Identificador único del indicador que será usado para definir el parámetro de alerta |
| alerta\_n1\_ind\_id | numeric(10,0) | | Identificador único del indicador que será usado para definir el parámetro de alerta |
| alerta\_min\_max | numeric(1,0) | | Parámetro de alerta con respecto a mínimo o máximo |
| alerta\_meta\_n1 | float | | Parámetro de alerta con respecto a la meta inferior |
| alerta\_meta\_n2 | float | | Parámetro de alerta con respecto a la meta superior |
| intranet | numeric(1,0) | | Campo utilizado para conversión entre versiones. Antes publicar el indicador en la intranet. |
| alerta\_n1\_tipo | numeric(1,0) | | Alerta verde indicadores mínimos o máximos |
| alerta\_n2\_tipo | numeric(1,0) | | Alerta amarilla indicadores mínimos o máximos |
| alerta\_n1\_fv | numeric(1,0) | | % de alerta verde |
| alerta\_n2\_fv | numeric(1,0) | | % de alerta amarilla |
| valor\_inicial\_cero | numeric(1,0) | | Flag que indica que el indicador debe comenzar en cero al inicio de cada año. |
| mascara\_decimales | numeric(1,0) | | Nro. de decimales que aplican al indicador |
| tipo\_medicion | numeric(1,0) | |  |
| url | varchar(500) | | URL asociada al indicador para efectos de la documentación del indicador |
| fecha \_ bloqueo \_ejecutado | datetime | | Fecha de bloqueo del registro de la data ejecutada del indicador |
| fecha\_bloqueo\_meta | datetime | | Fecha de bloqueo de la serie meta del indicador |
| psuperior\_id | numeric(10,0) | | Identificador único del indicador utilizado como parámetro de acotamiento superior |
| pinferior\_id | numeric(10,0) | | Identificador único del indicador utilizado como parámetro de acotamiento inferior |
| psuperior\_vf | float | | Parámetro superior |
| pinferior\_vf | float | | Parámetro inferior |
| multidimensional | numeric(1,0) | | Indica si el indicador es o no multidimensional |
| tipo | numeric(1,0) | | Campo utilizado para la migración a la nueva versión |
| tipo\_asociado | numeric(1,0) | | Campo utilizado para la migración a la nueva versión |
| asociado\_id | numeric(10,0) | | Identificador único del indicador tomado como base para la sumatoria o promedio |
| revision | numeric(3) | |  |
| ultimo\_programado | float | | Ultimo valor programado |
| tipo\_suma | numeric(1,0) | | Flag que indica si el indicador es del tipo sumatoria |
| asignar\_Inventario | numeric(1,0) | | Flag que indica si el indicador es de tipo INVENTARIO (aplica una fórmula de cálculo especial pare este tipo de indicador) |

|  |  |  |  |
| --- | --- | --- | --- |
| serie\_indicador | | Almacena las series de tiempo vinculadas a los indicadores. | |
| Attribute(s) of "serie\_indicador" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| serie\_id | numeric(10,0) | | Identificador único de la serie de tiempo |
| indicador\_id | numeric(10,0) | | Identificador único del indicador |
| naturaleza | numeric(1,0) | | Naturaleza de la serie (real, meta, programado, mínimo, máximo) |
| fecha\_bloqueo | datetime | | Fecha de bloqueo del registro |

|  |  |  |  |
| --- | --- | --- | --- |
| conjunto\_formula | | Almacena los ID y el tipo de serie (real, programado) de los indicadores que conforman a una fórmula de un indicador tipo FORMULA. | |
| Attribute(s) of "conjunto\_formula" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| padre\_id | numeric(10,0) | | Identificador único del indicador padre |
| serie\_id | numeric(10,0) | | Identificador único de la serie de tiempo |
| insumo\_serie\_id | numeric(10,0) | | Identificador único del identificador de la serie de tiempo. |
| indicador\_id | numeric(10,0) | | Identificador único del indicador al que pertenecen los insumos para la formula |

|  |  |  |  |
| --- | --- | --- | --- |
| duppont | | Almacena para cada usuario a indicador tipo formula, la configuración de los árboles de Dupont (número de niveles de profundidad y año-período por defecto). | |
| Attribute(s) of "duppont" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| indicador\_id | numeric(10) | | Identificador ID del indicador formula al cual está asociado el árbol de Dupont |
| usuario\_id | numeric(10) | | Identificador ID del usuario que creó la configuración |
| configuracion | varchar2(5000) | | Configuración del árbol (Año-Período desde y hasta y número de niveles del árbol) |

|  |  |  |  |
| --- | --- | --- | --- |
| medicion | | Almacena las Mediciones de los Indicadores para cada año y período. | |
| Attribute(s) of "medicion" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| serie\_id | numeric(10,0) | | Identificador único de la serie de tiempo vinculada a la medición (real, programado, mínimo, máximo) |
| indicador\_id | numeric(10,0) | | Identificador único del indicador |
| Ano | numeric(4,0) | | Año de la medición |
| Periodo | numeric(3,0) | | Número del período de la medición (mes, trimestre, etc.) |
| Valor | float | | Valor de la medición para el período y año del indicador y serie de tiempo |
| protegido | numeric(1,0) | | Flag que indica si la medición está protegida a fin de que no sea modificada por el usuario |

|  |  |  |  |
| --- | --- | --- | --- |
| explicacion | | Almacena las explicaciones (comentarios) que hacen los responsables de los indicadores e iniciativas (proyectos). | |
| Attribute(s) of "explicacion" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| explicacion\_id | numeric(10,0) | | Identificador único del registro asociado a la explicación |
| creado\_id | numeric(10,0) | | Identificador único del usuario que creó el registro |
| creado | datetime | | Fecha de creación del registro |
| objeto\_id | numeric(10,0) | | Identificador único del objeto al que corresponde la explicación (indicador, objetivo, iniciativa) |
| objeto\_key | numeric(2) | |  |
| titulo | varchar(250) | | Título de la explicación |
| fecha | datetime | | Fecha de la explicación |
| tipo | numeric(1) | | Tipo de explicación |
| fecha\_compromiso | datetime | |  |
| Fecha\_real | datetime | | Fecha de creación del registro (sistema) |

|  |  |  |  |
| --- | --- | --- | --- |
| explicacion\_adjunto | | Anexos (URL) de los anexos a las explicaciones de un objeto (indicador, iniciativa) | |
| Attribute(s) of "explicacion\_adjunto" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| explicacion\_id | numeric(10,0) | | Identificador único de la explicación |
| adjunto\_id | numeric(10,0) | | Identificador único del archivo adjunto |
| documento | text | | Descripción del documento anexo |
| mime\_type | varchar(50) | | Identificador único de la explicación |
| titulo | varchar(100) | | Descripción de la explicación |

|  |  |  |  |
| --- | --- | --- | --- |
| explicacion\_memo | | Almacena los campos tipo memo de una explicación vinculada a los indicadores o iniciativas. (Descripción de la desviación, causas de retrasos, estrategias de solución). | |
| Attribute(s) of "explicacion\_memo" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| explicacion\_id | numeric(10,0) | | Identificador único de la explicación |
| tipo | numeric(1,0) | | Tipo de explicación |
| memo | text | | Descripción de la explicación |

|  |  |  |  |
| --- | --- | --- | --- |
| formula | | Almacena la expresión que define a la fórmula de los indicadores de tipo FORMULA. | |
| Attribute(s) of "formula" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| indicador\_id | numeric(10,0) | | Identificador único del indicador |
| serie\_id | numeric(10,0) | | Identificador único de la serie de tiempo (real, programado, mínimo, máximo) |
| expresion | varchar(5000) | | Descripción detallada de la formula |

|  |  |  |  |
| --- | --- | --- | --- |
| grafico | | Almacena la configuración que hace un usuario sobre el gráfico de un indicador. (Colores de las series, año y periodo de graficación, nombre del gráfico, etc.) | |
| Attribute(s) of "grafico" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| grafico\_id | numeric(10,0) | | Identificador único del registro asociado al gráfico |
| organizacion\_id | numeric(10,0) | | Identificador único de la organización a la que pertenece el grafico |
| nombre | varchar(100) | | Nombre del gráfico |
| configuracion | varchar(5000) | | Configuración del gráfico (año y período desde hasta, tipo de gráfico, color de las series de tiempo, indicadores que conforman al gráfico, etc.) |
| usuario\_id | numeric(10,0) | | Identificador único del usuario creó el gráfico |
| objeto\_id | numeric(10,0) | | Identificador único del indicador en el gráfico |
| className | varchar(50) | | Identificador único de la clase de Java |

|  |  |  |  |
| --- | --- | --- | --- |
| categoria\_por\_indicador | | Almacena las categorías o expresiones utilizadas en la definición de indicadores cualitativos | |
| Attribute(s) of "categoria\_por\_indicador" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| indicador\_id | numeric(10,0) | | Identificador único del indicador de tipo cualitativo |
| categoria\_id | numeric(10,0) | | Identificador único de la categoría que aplica al indicador cualitativo |
| orden | numeric(4,0) | | Orden que ocupa la expresión lingüística en la escala de medición del indicador cualitativo ordinal |

**Módulo de Planes:**

|  |  |  |  |
| --- | --- | --- | --- |
| planes | | Almacena los datos básicos de los planes estratégicos y operativos. | |
| Attribute(s) of "planes" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| plan\_id | numeric(10,0) | | Identificador único del Plan |
| padre\_id | numeric(10,0) | | Identificador único del plan padre |
| plan\_impacta\_id | numeric(10,0) | | Identificador único del plan relacionado |
| organizacion\_id | numeric(10,0) | | Identificador único de la organización a la que pertenece el plan |
| nombre | varchar(50) | | Nombre del plan |
| ano\_inicial | numeric(4,0) | | Año de inicio del plan |
| ano\_final | numeric(4,0) | | Año de culminación |
| Tipo | numeric(1,0) | | Tipo de plan (estratégico, operativo) |
| activo | numeric(1,0) | | Flag que indica que el plan está activo o no |
| revision | numeric(2,0) | | Nro. de revisión o copia del plan |
| metodologia\_id | numeric(10,0) | | Identificador único de la plantilla o metodología de planeación (plantilla de planes) empleada para construir el plan |
| clase\_id | numeric(10,0) | | Identificador único de la clase de indicadores donde serán almacenados los indicadores de logro parciales y anuales de los objetivos y perspectivas del plan. |
| Clase \_ id \_ indicadores\_totales | numeric(10,0) | | Identificador único de la clase de indicadores donde serán almacenados los indicadores de logro consolidados del plan. |
| ind\_total\_imputacion \_id | numeric(10,0) | | Identificador único del indicador que almacenará el presupuesto consolidado de las iniciativas del plan. |
| ind\_total\_iniciativa\_id | numeric(10,0) | | Identificador único del indicador que almacenará el valor ejecutado consolidado de las iniciativas del plan. |
| ultima\_medicion\_anual | float | | Valor de la última medición del % de logro anual del indicador |
| Ultima\_medición \_parcial | float | | Valor de la última medición del % de logro parcial del indicador |
| nl\_ano\_indicador\_id | numeric(10,0) | | Identificador único del indicador que almacena el % de logro anual del plan |
| nl\_par\_indicador\_id | numeric(10,0) | | Identificador único del indicador que almacena el % de logro parcial del plan |
| fecha\_ultima\_medicion | varchar(10) | | Fecha de la última medición de los indicadores de logro del plan |
| crecimiento | numeric(1,0) | |  |
| esquema | varchar(50) | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| indicador\_por\_plan | | Almacena los ID de los indicadores asociados a una perspectiva (dimensión) de un plan, así como el peso (importancia) del indicador. | |
| Attribute(s) of "indicador\_por\_plan" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| plan\_id | numeric(10,0) | | Identificador único del plan |
| indicador\_id | numeric(10,0) | | Identificador único del indicador de logro asignado al plan |
| peso | float | | Peso asignado al indicador |
| crecimiento | numeric(1) | |  |
| tipo\_medicion | numeric(1) | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| perspectiva | | Almacena las Perspectivas (dimensiones estratégicas) de un plan. | |
| Attribute(s) of "perspectiva" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| perspectiva\_id | numeric(10,0) | | Identificador único de la perspectiva o dimensión estratégica |
| padre\_id | numeric(10,0) | | Identificador único de la perspectiva padre de la perspectiva que se está creando (árbol) |
| plan\_id | numeric(10,0) | | Identificador único del plan al que pertenece la perspectiva |
| responsable\_id | numeric(10,0) | | Identificador único del responsable de la perspectiva |
| nombre | varchar(250) | | Nombre de la perspectiva |
| descripcion | varchar(2000) | | Descripción de la perspectiva |
| orden | numeric(2,0) | | Orden que ocupa la perspectiva dentro del plan |
| frecuencia | numeric(1,0) | | Frecuencia de evaluación (trimestral, mensual, etc.) |
| titulo | varchar(50) | | Nombre del tipo de elemento |
| tipo | numeric(1,0) | | Tipo de elemento (perspectiva, objetivo, etc.) |
| fecha\_ultima\_medicion | varchar(10) | | Fecha de la última medición |
| ultima\_medicion\_anual | float | | Valor de la última medición anual |
| ultima\_medicion\_parcial | float | | Valor de la última medición parcial |
| creado | datetime | | Fecha de creación del registro |
| modificado | datetime | | Fecha de creación del registro |
| creado\_id | numeric(10,0) | | Identificador único del usuario que creó el registro |
| modificado\_id | numeric(10,0) | | Identificador único del usuario que modificó el registro |
| ano | numeric(10,0) | | Año |
| crecimiento\_parcial | numeric(1,0) | |  |
| crecimiento\_anual | numeric(1,0) | |  |
| clase\_id | numeric(10,0) | | Identificador único de la clase de indicadores en la que se almacenan los indicadores de logro de la perspectiva |
| tipo\_calculo | numeric(1,0) | | Flag que indica si el sistema generará automáticamente los indicadores de logro de la perspectiva |
| nl\_par\_indicador\_id | numeric(10,0) | | Identificador único del indicador de logro parcial de la perspectiva |
| nl\_ano\_indicador\_id | numeric(10,0) | | Identificador único del indicador de logro anual de la perspectiva |

|  |  |  |  |
| --- | --- | --- | --- |
| perspectiva\_nivel | | Almacena los porcentajes de logro de las perspectivas (dimensiones) de un plan para cada año y período de vigencia del plan. | |
| Attribute(s) of "perspectiva\_nivel" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| perspectiva\_id | numeric(10,0) | | Identificador único de la perspectiva o dimensión estratégica |
| Tipo | numeric(1,0) | | Tipo de elemento (perspectiva, objetivo, etc.) |
| Ano | numeric(4,0) | | Año |
| periodo | numeric(3,0) | | Período (trimestre, mes, etc.) |
| Nivel | float | | Nivel del elemento dentro del árbol de dimensiones estratégicas (perspectivas) |

|  |  |  |  |
| --- | --- | --- | --- |
| perspectiva\_relacion | | Almacena el orden de las perspectivas (dimensiones) de un plan. | |
| Attribute(s) of "perspectiva\_relacion" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| perspectiva\_id | numeric(10,0) | | Identificador único de la perspectiva o dimensión estratégica |
| relacion\_id | numeric(10,0) | | Almacena el objeto relacionado de otro plan |

|  |  |  |  |
| --- | --- | --- | --- |
| plan\_nivel | | Almacena los porcentajes de logro consolidado de un plan para cada año y período de vigencia del plan. | |
| Attribute(s) of "plan\_nivel" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| plan\_id | numeric(10,0) | | Identificador único del Plan |
| Tipo | numeric(1,0) | | Tipo de elemento (perspectiva, objetivo, etc.) |
| Ano | numeric(4,0) | | Año |
| Periodo | numeric(3,0) | | Período (trimestre, mes, etc.) |
| Nivel | float | | Número del nivel dentro del árbol de dimensiones del plan |

|  |  |  |  |
| --- | --- | --- | --- |
| indicador\_por\_perspectiva | | Almacena los ID de los indicadores asociados a una perspectiva (dimensión) de un plan | |
| Attribute(s) of "indicador\_por\_perspectiva" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| perspectiva\_id | numeric(10,0) | | Identificador único de la perspectiva del plan |
| indicador\_id | numeric(10,0) | | Identificador único del indicador de logro asociado a la perspectiva |
| peso | float | | Peso asignado a la perspectiva |

|  |  |  |  |
| --- | --- | --- | --- |
| meta | | Registro de las Metas de los Indicadores asociados a un Plan | |
| Attribute(s) of "meta" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| plan\_id | numeric(10,0) | | Identificador único del plan |
| indicador\_id | numeric(10,0) | | Identificador único del indicador |
| tipo | numeric(1,0) | | Tipo de meta (parcial o anual) |
| ano | numeric(4,0) | | Año al que corresponde la meta |
| periodo | numeric(3,0) | | Número del período (mes, trimestre, etc.) a la que corresponde la medición. Depende de la frecuencia con que fue definido el indicador |
| serie\_id | numeric(10,0) | | Identificador único de la serie de tiempo (real, meta, programado, mínimo o máximo) |
| valor | float | | Valor de la meta para el Plan e Indicador |
| protegido | numeric(1,0) | | Flag que indica si la medición está protegida |

|  |  |  |  |
| --- | --- | --- | --- |
| modelo | | Almacena los datos básicos de los modelos o mapas causa efecto creados por los usuarios. | |
| Attribute(s) of "modelo" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| modelo\_id | numeric(10,0) | | Identificador único del registro |
| plan\_id | numeric(10,0) | | Identificador único del Plan |
| Descripción | varchar(2000) | | Descripción del Modelo o Mapa Causa Efecto |
| binario | text | | Campo donde se almacena la gráfica del modelo o mapa causa efecto |

|  |  |  |  |
| --- | --- | --- | --- |
| indicador\_estado | | Almacena los porcentajes de logro parcial y anual (estado) de los indicadores asociados a un plan. | |
| Attribute(s) of "indicador\_estado" Entity | | | | |
| **Name** | **Datatype** | | **Definition** | |
| plan\_id | numeric(10,0) | | Identificador único del Plan al que pertenece el indicador | |
| indicador\_id | numeric(10,0) | | Identificador único del Indicador | |
| tipo | numeric(1,0) | | Tipo de Estado del indicador 0 = Anual, 1= Parcial | |
| ano | numeric(4,0) | | Año | |
| periodo | numeric(3,0) | | Período (trimestre, mes, etc.) | |
| estado | float | | Valor del estado (porcentaje de logro del indicador) para el año y período | |

**Módulo de Iniciativas:**

|  |  |  |  |
| --- | --- | --- | --- |
| iniciativa | | Almacena los Datos Básicos de las Iniciativas (proyectos) | |
| Attribute(s) of "iniciativa" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| iniciativa\_id | numeric(10,0) | | Identificador único de la Iniciativa |
| organizacion\_id | numeric(10,0) | | Identificador único de la organización a la que pertenece la iniciativa |
| nombre | varchar(250) | | Nombre de la iniciativa |
| nombre\_largo | varchar(300) | | Nombre largo de la iniciativa |
| creado | datetime | | Fecha de creación del registro |
| alerta\_iniciativa\_zv | numeric(4,0) | | % de zona verde |
| alerta\_iniciativa\_za | numeric(4,0) | | % de zona amarilla |
| frecuencia | numeric(2,0) | | Frecuencia de medición |
| tipo\_alerta | numeric(2,0) | | Tipo de cálculo de avance. Automático o manual. (ya no aplica) |
| contratista | varchar(50) | | Descripción del ejecutor de la iniciativa (en caso de ser un ente externo como una constratista) |
| iniciativa\_asociada\_id | numeric(10,0) | | En caso de ser una iniciativa que viene derivada de otra, este campo define el identificador de la iniciativa asociada. |
| iniciativa\_asociada\_plan\_id | numeric(10,0) | | En caso de ser una iniciativa que viene derivada de otra, este campo define el identificador del plan al que pertenece la iniciativa asociada. |
| naturaleza | numeric(1,0) | | Descripción de la naturaleza de la iniciativa |
| resp\_fijar\_meta\_id | numeric(10,0) | | Identificador único del responsable de fijar la meta |
| resp\_lograr\_meta\_id | numeric(10,0) | | Identificador único del responsable de lograr la meta |
| resp\_seguimiento\_id | numeric(10,0) | | Identificador único del responsable de seguimiento de la iniciativa |
| resp\_cargar\_meta\_id | numeric(10,0) | | Identificador único del responsable de cargar la meta |
| resp\_cargar\_ejecutado\_id | numeric(10,0) | | Identificador único del responsable de cargar la data ejecutada |
| visible | numeric(1,0) | | Flag que indica si el proyecto es o no visible en el visor de iniciativas y en el visor de planes. Se oculta cuando las iniciativas pasan a historico |
| clase\_id | numeric(10,0) | | Identificador único de la clase de indicadores donde serán almacenados los indicadores tanto de avance como de gestión de la iniciativa |
| read\_only | numeric(1,0) | | Flag de solo lectura del registro |
| porcentaje\_completado | float | | % de ejecución de la iniciativa |
| crecimiento | numeric(1,0) | |  |
| fecha\_ultima\_medicion | datetime | | Fecha de la última medición |
| tipo\_medicion | numeric(1,0) | |  |
| historico\_date | datetime | | Fecha de pase al archivo histórico de la iniciativa |
| estatusId | numeric(10) | | Identificador ID del estatus actual de la iniciativa |

|  |  |  |  |
| --- | --- | --- | --- |
| iniciativa\_ano | | Almacena el porcentaje de avance al cierre de cada año para una iniciativa. | |
| Attribute(s) of "iniciativa\_ano" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| iniciativa\_id | numeric(10,0) | | Identificador único de la iniciativa |
| ano | numeric(4,0) | | Año |
| resultado | text | | Descripción de resultado esperado de la iniciativa para el año indicado |

|  |  |  |  |
| --- | --- | --- | --- |
| iniciativa\_plan | | Almacena los ID de las iniciativas que están asociadas a un Plan. | |
| Attribute(s) of "iniciativa\_plan" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| iniciativa\_id | numeric(10,0) | | Identificador único de la iniciativa |
| plan\_id | numeric(10,0) | | Identificador único del plan al que está asociado la iniciativa |

|  |  |  |  |
| --- | --- | --- | --- |
| iniciativa\_por\_perspectiva | | Almacena los ID de las iniciativas vinculadas a una perspectiva (dimensión) de un plan. | |
| Attribute(s) of "iniciativa\_por\_perspectiva" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| perspectiva\_id | numeric(10,0) | | Identificador único de la perspectiva del plan a la que está asociada la iniciativa |
| iniciativa\_id | numeric(10,0) | | Identificador único de la iniciativa |

|  |  |  |  |
| --- | --- | --- | --- |
| inc\_actividad | | Almacena las Actividades de una Iniciativa (proyecto). | |
| Attribute(s) of "inc\_actividad" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| actividad\_id | numeric(10,0) | | Identificador único de la actividad |
| producto\_esperado | varchar(2000) | | Descripción del producto que se espera obtener al finalizar la ejecución de la actividad |
| recurso\_humano | char varying(2000) | | Descripción del recurso humano que se espera utilizar en la ejecución de la actividad |
| recurso\_material | char varying(2000) | | Descripción del recurso material que se espera utilizar en la ejecución de la actividad |
| peso | float | | Peso de la actividad. Utilizado para calcular el avance de la iniciativa (proyecto) |
| alerta\_za | float | | % de desviación con respecto a lo programado para indicar que el indicador está en alerta amarilla. |
| alerta\_zv | float | | % de desviación con respecto a lo programado para indicar que el indicador está en alerta verde. |
| porcentaje\_ejecutado | float | | % de ejecución real de la actividad |
| porcentaje\_esperado | float | | % de ejecución esperado (programado) de la actividad |
| porcentaje\_completado | float | | % de ejecución completado de la actividad |
| alerta | numeric(4,0) | | Color de la alerta de la actividad (rojo, verde, amarillo o blanco) |

|  |  |  |  |
| --- | --- | --- | --- |
| indicador\_por\_iniciativa | | Almacena los ID de los indicadores que están asociados a una iniciativa (Avance, Presupuesto, Eficiencia, Eficacia y otros definidos por los usuarios) | |
| Attribute(s) of "indicador\_por\_iniciativa" Entity | | | | |
| **Name** | **Datatype** | | **Definition** | |
| iniciativa\_id | numeric(10,0) | | Identificador único de la Iniciativa (proyecto) | |
| indicador\_id | numeric(10,0) | | Identificador único del Indicador asociado a la iniciativa | |
| tipo | numeric(1,0) | | Tipo de Estado del indicador 0 = Anual, 1= Parcial | |

|  |  |  |  |
| --- | --- | --- | --- |
| pry\_actividad | | Almacena las actividades de una iniciativa.  Hay dos tablas que almacenan actividades de la iniciativa OJO | |
| Attribute(s) of "pry\_actividad" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| actividad\_id | numeric(10,0) | | Identificador único de la Actividad |
| padre\_id | numeric(10,0) | | Identificador único de la actividad padre |
| proyecto\_id | numeric(10,0) | | Identificador único del proyecto |
| nombre | varchar(250) | | Nombre de la Actividad |
| descripcion | varchar(5000) | | Descripción de la Actividad |
| comienzo\_plan | datetime | | Fecha de comienzo planificada |
| comienzo\_real | datetime | | Fecha de inicio real de la actividad |
| fin\_plan | datetime | | Fecha planificada de culminación |
| fin\_real | datetime | | Fecha de culminación real |
| duracion\_plan | float | | Duración en días planificada |
| duracion\_real | float | | Duración en días real ejecutada |
| unidad\_id | numeric(10,0) | | Identificador único de la unidad de medida del presupuesto del proyecto (USD, bolívares, pesos) |
| fila | numeric(10,0) | | Nro. de fila ubicación de la actividad en el proyecto |
| nivel | numeric(4,0) | | Nro. del nivel que ocupa la actividad en el proyecto |
| compuesta | numeric(1,0) | | Flag que indica si la actividad es simple o compuesta |
| creado | datetime | | Fecha de creación del registro |
| modificado | datetime | | Fecha de creación del registro |
| creado\_id | numeric(10,0) | | Identificador único del usuario que creó la actividad |
| modificado\_id | numeric(10,0) | | Identificador único del usuario que modificó la actividad |
| indicador\_id | numeric(10,0) | | Identificador único del indicador de avance de la actividad |
| objeto\_asociado\_id | numeric(10,0) | | Identifica el proyecto o actividad de otro plan vinculada a la actividad que se está creando |
| naturaleza | numeric(1,0) | | Naturaleza de la actividad (simple o formula) |
| clase\_id | numeric(10,0) | | Identificador único de la clase de indicadores que almacena el indicador de avance de la actividad |
| iniciativa\_asociada\_plan\_id | numeric(10,0) | | Identificador único de la iniciativa vinculada |
| resp\_fijar\_meta\_id | numeric(10,0) | | Identificador único del responsable de fijar la meta |
| resp\_lograr\_meta\_id | numeric(10,0) | | Identificador único del responsable de lograr la meta |
| resp\_seguimiento\_id | numeric(10,0) | | Identificador único del responsable de hacer el seguimiento de la actividad |
| resp\_cargar\_ejecutado\_id | numeric(10,0) | | Identificador único del responsable de cargar la data ejecutada de la actividad |
| resp\_cargar\_meta\_id | numeric(10,0) | | Identificador único del responsable de cargar la data meta o programado de la actividad |
| tipo\_medicion | numeric(1,0) | | Tipo de medición para la actividad manual o automática |
| objeto\_asociado\_plan\_id | numeric(10,0) | | Identifica el identificador único del proyecto o actividad de otro plan vinculada a la actividad que se está creando |
| Objeto \_ asociado \_className | varchar(50) | | Identifica el nombre de la clase del proyecto o actividad de otro plan vinculada a la actividad que se está creando |
| crecimiento | numeric(1,0) | |  |
| porcentaje\_completado | float | | Porcentaje de avance de la actividad |
| porcentaje\_esperado | float | | Porcentaje de avance programado (esperado) |
| porcentaje\_ejecutado | float | | Porcentaje de avance en el período |
| fecha\_ultima\_medicion | datetime | | Fecha de la última medición registrada |

|  |  |  |  |
| --- | --- | --- | --- |
| pry\_calendario | | Almacena los días laborables para la determinación de las fechas de inicio y fin de las iniciativas. | |
| Attribute(s) of "pry\_calendario" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| calendario\_id | numeric(10,0) | | Identificador único del tipo de calendario del proyecto |
| nombre | varchar(50) | | Nombre del calendario |
| Dom | numeric(1,0) | | Flag de día domingo laborable |
| Lun | numeric(1,0) | | Flag de lunes laborable |
| Mar | numeric(1,0) | | Flag de martes laborable |
| Mie | numeric(1,0) | | Flag de miércoles laborable |
| Jue | numeric(1,0) | | Flag de jueves laborable |
| Vie | numeric(1,0) | | Flag de viernes laborable |
| Sab | numeric(1,0) | | Flag de sabado laborable |
| proyecto\_id | numeric(10,0) | | Identificador único del proyecto al cual está asociado el calendario |

|  |  |  |  |
| --- | --- | --- | --- |
| pry\_calendario\_detalle | | Almacena los días no laborables empleados en el calendario de las iniciativas. | |
| Attribute(s) of "pry\_calendario\_detalle" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| calendario\_id | numeric(10,0) | | Identificador único del tipo de calendario del proyecto |
| fecha | datetime | | Fecha |
| feriado | numeric(1,0) | | Flag que indica si la fecha anterior es laborable |

**Módulo de Panel de Control:**

|  |  |  |  |
| --- | --- | --- | --- |
| vista | | Almacena los datos básicos de los Paneles de control (Presentaciones Ejecutivas). | |
| Attribute(s) of "vista" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| vista\_id | numeric(10,0) | | Identificador único de la vista |
| organizacion\_id | numeric(10,0) | | Identificador único de la organización a la que pertenece la vista |
| nombre | varchar(50) | | Nombre de la vista |
| descripcion | text | | Descripción de la vista |
| visible | numeric(1,0) | | Flag que indica si la vista es visible o no |
| fecha\_inicio | varchar(10) | | Fecha de inicio, marco de referencia para generación de los gráficos de la vista |
| fecha\_fin | varchar(10) | | Fecha de culminación |

|  |  |  |  |
| --- | --- | --- | --- |
| pagina | | Almacena los datos básicos de las Páginas que conforman a de un Panel de Control (presentación ejecutiva). Nombre de la página, ancho y alto de las celdas, etc. | |
| Attribute(s) of "pagina" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| pagina\_id | numeric(10,0) | | Identificador único de la página de la presentación ejecutiva |
| vista\_id | numeric(10,0) | | Identificador único de la vista de la presentación ejecutiva |
| descripcion | varchar(2000) | | Descripción de la vista |
| filas | numeric(2,0) | | Número de filas de la página |
| columnas | numeric(2,0) | | Número de columnas de la página |
| ancho | numeric(4,0) | | Ancho en pixeles de la celda |
| alto | numeric(4,0) | | Alto en pixeles de la celda |
| numero | numeric(4,0) | | Cantidad de celdas en la página |
| portafolioId | numeric(10,0) | | Identificador único del portafolio de proyectos al cual está asociada la página. |

|  |  |  |  |
| --- | --- | --- | --- |
| celda | | Celdas de una Presentación Ejecutiva (panel de control) | |
| Attribute(s) of "celda" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| celda\_id | numeric(10,0) | | Identificador único del registro |
| pagina\_id | numeric(10,0) | | Identificador único del número de la página del panel de control (presentación ejecutiva) |
| titulo | varchar(50) | | Título de la celda |
| fila | numeric(2,0) | | Tamaño en pixeles de las filas de la celda |
| columna | numeric(2,0) | | Tamaño en pixeles de las columnas de la celda |
| configuracion | longvarchar | | Indica si muestra la alerta, árbol de dupont en la celda |

**Módulo de Visor de Datos:**

|  |  |  |  |
| --- | --- | --- | --- |
| reporte | | Almacena los datos básicos de los reportes creados por los usuarios en el módulo de VISTAS DE DATOS. | |
| Attribute(s) of "reporte" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| reporte\_id | numeric(10,0) | | Identificador único del reporte |
| organizacion\_id | numeric(10,0) | | Identificador único de la organización a la que pertenece el reporte |
| nombre | varchar(50) | | Nombre del reporte |
| descripcion | varchar(1000) | | Descripción del contenido del reporte |
| publico | numeric(1) | | Flag que indica si el reporte es público (puede ser visto por todos los usuarios que tengan acceso a la organización) o solo puede ser visto por el usuario que creo el reporte |
| tipo | numeric(1) | | Indica el formato del reporte, longitudinal o transversal. |
| corte | numeric(1) | |  |
| configuracion | varchar(5000) | | Configuración del reporte. Indicadores seleccionados, Organizaciones seleccionadas, variables a ser incorporadas al reporte, año y periodo desde y hasta de ejecución |
| usuario\_id | numeric(10,0) | | Identificador único del usuario que creó el reporte |

**Módulo de Portafolio:**

|  |  |  |  |
| --- | --- | --- | --- |
| portafolio | | Almacena los datos básicos de los portafolios de proyectos. | |
| Attribute(s) of "portafolio" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| id | numeric(10,0) | | Identificador único del Portafolio |
| clase\_id | numeric(10,0) | | Identificador único de la clase de indicadores donde será almacenado el indicador de avance físico del portafolio |
| organizacion\_id | numeric(10,0) | | Identificador único de la organización a la que pertenece el portafolio |
| Nombre | varchar(50) | | Nombre del Portafolio de Proyectos |
| Activo | numeric(1) | | Flag que indica si el portafolio está activo o en histórico |
| porcentaje\_completado | float | | Porcentaje de avance de los proyectos que están asociados al portafolio |
| estatusid | numeric(10,0) | | Identificador único del estatus actual de portafolio |
| frecuencia | numeric(1) | | Frecuencia de evaluación del portafolio (mensual, trimestral, etc.) |
| fecha\_ultimo\_calculo | datetime | | Fecha del último cálculo del avance consolidado del portafolio |

|  |  |  |  |
| --- | --- | --- | --- |
| portafolio\_iniciativa | | Almacena los ID y el peso (importancia) de las iniciativas que están asociadas a un portafolio. | |
| Attribute(s) of "portafolio\_iniciativa" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| portafolio\_id | numeric(10,0) | | Identificador único del Portafolio |
| iniciativa\_id | numeric(10,0) | | Identificador único de la iniciativa (proyecto) que está asociada al portafolio |
| peso | float | | Peso de la iniciativa (proyecto) dentro del portafolio |

|  |  |  |  |
| --- | --- | --- | --- |
| indicador\_por\_portafolio | | Almacena los ID de los indicadores asociados a un Portafolio de Iniciativas. (Avance del portafolio) | |
| Attribute(s) of "indicador\_por\_portafolio" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| indicador\_id | numeric(10,0) | | Identificador único del indicador de avance del portafolio |
| portafolio\_id | numeric(10,0) | | Identificador único del portafolio al cual está asociado el indicador de avance |
| tipo | numeric(1) | |  |

**Módulo de Problemas (Acciones Correctivas):**

|  |  |  |  |
| --- | --- | --- | --- |
| clase\_problema | | Almacena los grupos o clases de problemas | |
| Attribute(s) of "clase\_problema" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| clase\_id | numeric(10,0) | | Identificador único de la Clase de Problemas |
| padre\_id | numeric(10,0) | | Identificador único de la Clase Padre (árbol) de problemas |
| organizacion\_id | numeric(10,0) | | Identificador único de la Organización a la que pertenece la Clase de Problema |
| nombre | varchar(50) | | Nombre de la Clase |
| descripcion | varchar(2000) | | Descripción de la Clase |
| creado | datetime | | Fecha de creación del registro |
| modificado | datetime | | Fecha de creación del registro |
| creado\_id | numeric(10,0) | | Identificador único del usuario que creó el registro |
| modificado\_id | numeric(10,0) | | Identificador único del usuario que modificó el registro |
| tipo | number(1) | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| problema | | Almacena los datos básicos de los problemas. Módulo de Acciones Correctivas. | |
| Attribute(s) of "problema" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| problema\_id | numeric(10,0) | | Identificador único del Problema |
| organizacion\_id | numeric(10,0) | | Identificador único de la organización a la que pertenece el problema |
| indicador\_efecto\_id | numeric(10,0) | | Identificador único del indicador mediante el cual se detectó el problema |
| iniciativa\_efecto\_id | numeric(10,0) | | Identificador único de la iniciativa mediante el cual se detectó el problema |
| fecha | datetime | | Fecha de formulación del problema |
| responsable\_id | numeric(10,0) | | Identificador único del responsable de formular el problema |
| auxiliar\_id | numeric(10,0) | | Identificador único del responsable auxiliar de formular el problema |
| estado\_id | numeric(10,0) | | Identificador del estado actual del problema (sin iniciar, en ejecución, avanzado, etc.) |
| fecha\_estimada\_inicio | datetime | | Fecha estimada de inicio de las acciones correctivas del problema |
| fecha\_estimada\_final | datetime | | Fecha estimada de culminación de las acciones correctivas del problema |
| fecha\_real\_inicio | datetime | | Fecha real de inicio de las acciones correctivas del problema |
| fecha\_real\_final | datetime | | Fecha real de culminación de las acciones correctivas del problema |
| creado | datetime | | Fecha de creación del registro |
| modificado | datetime | | Fecha de creación del registro |
| creado\_id | numeric(10,0) | | Identificador único del usuario que creó el problema |
| modificado\_id | numeric(10,0) | | Identificador único del usuario que modificó el problema |
| nombre | varchar(50) | | Descripción del problema |
| read\_only | numeric(1,0) | | Flag que indica si el registro es de solo lectura |
| indicador\_id | numeric(10,0) | | Identificador único del indicador mediante el cual se detectó el problema |
| iniciativa\_id | numeric(10,0) | | Identificador único de la iniciativa mediante la cual se detectó el problema |
| clase\_id | numeric(10,0) | | Identificador único de la clase donde se almacenó el problema |

|  |  |  |  |
| --- | --- | --- | --- |
| problema\_memo | | Almacena los campos de descripción de un Problema. Módulo de Acciones Correctivas. | |
| Attribute(s) of "problema\_memo" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| problema\_id | numeric(10,0) | | Identificador único del problema |
| Tipo | numeric(1,0) | | Tipo de Campo (Descripción, Causas, Estrategias de Solución) |
| memo | text | | Descripción del Campo |

|  |  |  |  |
| --- | --- | --- | --- |
| causa\_por\_problema | | Causas asociadas a un problema en particular | |
| Attribute(s) of "causa\_por\_problema” | | | |
| **Name** | **Datatype** | | **Definition** |
| problema\_id | numeric(10,0) | | Identificador único del Problema |
| causa\_id | numeric(10,0) | | Identificador único de la Causa (tabla de causas) |
| orden | numeric(5,0) | | Orden de ingreso de la causa en el problema |

|  |  |  |  |
| --- | --- | --- | --- |
| accion | | Acciones Correctivas asociadas a un problema | |
| Attribute(s) of "accion" Entity | | | |
| **Atributo** | **Tipo de Dato** | | **Descripción** |
| accion\_id | numeric(10,0) | | Identificador único de la Acción Correctiva |
| padre\_id | numeric(10,0) | | Identificador único de la Acción Correctiva padre |
| problema\_id | numeric(10,0) | | Identificador único del problema al que pertenece la acción correctiva |
| estado\_id | numeric(10,0) | | Identificador único del estatus de la acción |
| nombre | varchar(50) | | Nombre de la Acción |
| fecha\_estimada\_inicio | datetime | | Fecha estimada de inicio de la acción |
| fecha\_estimada\_final | datetime | | Fecha estimada de culminación de la acción |
| fecha\_real\_inicio | datetime | | Fecha real de inicio de la acción |
| fecha\_real\_final | datetime | | Fecha real de culminación de la acción |
| frecuencia | numeric(5,0) | | Frecuencia de seguimiento en días |
| orden | numeric(2,0) | | Orden que ocupa la acción en la lista de acciones asociadas al problema |
| creado | datetime | | Fecha de creación del registro en la BD |
| modificado | datetime | | Fecha de modificación del registro |
| creado\_id | numeric(10,0) | | Identificador único del usuario que creó el registro |
| modificado\_id | numeric(10,0) | | Identificador único del usuario que modificó el registro |
| descripcion | text | | Descripción del Problema |
| read\_only | numeric(1,0) | | Flag que indica si el registro es de solo lectura |

|  |  |  |  |
| --- | --- | --- | --- |
| responsable\_por\_accion | | Almacena los ID de los responsables de las acciones correctivas vinculadas a un problema. | |
| Attribute(s) of "responsable\_por\_accion" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| accion\_id | numeric(10,0) | | Identificador único de la Acción Correctiva |
| responsable\_id | numeric(10,0) | | Identificador único del responsable de la acción |
| tipo | numeric(1,0) | | Tipo de responsable (lograr meta, hacer seguimiento, ejecutar, etc.) |

|  |  |  |  |
| --- | --- | --- | --- |
| seguimiento | | Almacena los reportes de seguimiento a las acciones correctivas asociadas a un problema. | |
| Attribute(s) of "seguimiento" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| seguimiento\_id | numeric(10,0) | | Identificador único del reporte de seguimiento |
| accion\_id | numeric(10,0) | | Identificador único de la Acción Correctiva a la que se le está haciendo seguimiento |
| estado\_id | numeric(10,0) | | Identificador único del estatus de la acción correctiva |
| fecha\_emision | datetime | | Fecha de emisión del reporte de seguimiento |
| emision\_enviado | numeric(1,0) | | Fecha de envío del mensaje al responsable de reportar avance de la acción correctiva |
| fecha\_recepcion | datetime | | Fecha de recepción de la solicitud de seguimiento |
| recepcion\_enviado | numeric(1,0) | | Fecha de envío del reporte de seguimiento por parte del responsable |
| fecha\_aprobacion | datetime | | Fecha en que el supervisor de la acción correctiva aprobó el reporte de seguimiento |
| preparado\_por | varchar(50) | | Descripción de quién preparó el reporte |
| numero\_de\_reporte | numeric(5,0) | | Nro. de reporte de seguimiento |
| aprobado | numeric(1,0) | | Flag que indica si fue aprobado o no el reporte |
| aprobado\_por | varchar(50) | | Descripción de quien aprobó el reporte |
| clave\_correo | varchar(50) | | Correo electrónico del responsable |
| creado | datetime | | Fecha de creación del registro |
| modificado | datetime | | Fecha de modificación del registro |
| creado\_id | numeric(10,0) | | Identificador único del usuario que creó el reporte de seguimiento |
| modificado\_id | numeric(10,0) | | Identificador del usuario que modificó el reporte |
| read\_only | numeric(1,0) | | Flag de solo lectura, no modificable |

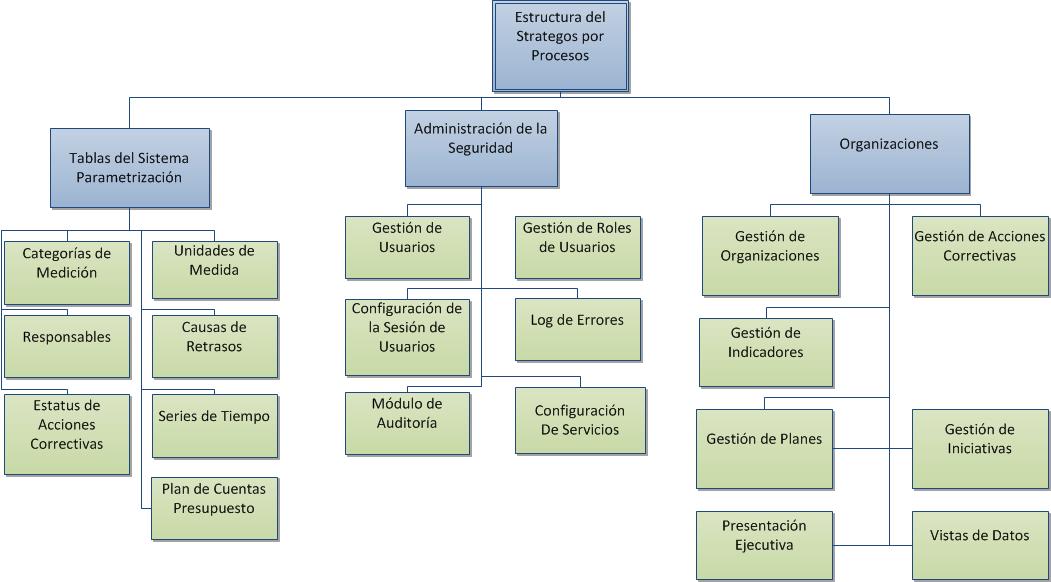
|  |  |  |  |
| --- | --- | --- | --- |
| seguimiento\_memo | | Almacena los campos memos de un reporte de seguimiento asociado a una acción correctiva de un problema. | |
| Attribute(s) of "seguimiento\_memo" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| seguimiento\_id | numeric(10,0) | | Identificador único del reporte de seguimiento |
| Tipo | numeric(1,0) | | Tipo de Seguimiento |
| memo | text | | Descripción del reporte de seguimiento |

|  |  |  |  |
| --- | --- | --- | --- |
| seguimiento\_mensaje\_email | | Almacena la configuración del mensaje de correo electrónico que será enviado a los responsables de ejecutar una acción correctiva vinculada a un problema. | |
| Attribute(s) of "seguimiento\_mensaje\_email" Entity | | | |
| **Name** | **Datatype** | | **Definition** |
| mensaje\_id | numeric(10,0) | | Identificador único del mensaje |
| descripcion | varchar(50) | | Descripción del mensaje |
| mensaje | text | | Texto del mensaje |
| acuse\_recibo | numeric(1,0) | | Flag que indica que el responsable recibió el mensaje |
| solo\_auxiliar | numeric(1,0) | | Flag que indica que el mensaje fue enviado al auxiliar de seguimiento |
| creado | datetime | | Fecha de creación del registro |
| modificado | datetime | | Fecha de modificación del registro |
| creado\_id | numeric(10,0) | | Identificador único del usuario que creo el registro |
| modificado\_id | numeric(10,0) | | Identificador único del usuario que modificó el registro |

# CAPITULO II

## Arquitectura de Aplicación

### Estructura del Strategos Advance



### Requerimientos de Hardware y Software Comercial

**Configuración Recomendada Servidor WEB:**

|  |  |
| --- | --- |
| Número Máximo de Usuarios | Características Hardware / Comentarios |
| 20 | Procesador Xeon– 2.4 GHz 12 MB Cache  1 Quad Core CPU  16 GB Ram  Partición 100 GB para OS |
| 21 y 99 | Procesador Xeon– 2.4 GHz 12 Cache  2 Quad Core CPU  32 GB Ram  Partición de 100 GB para OS |
| Más de 100 usuarios | Procesador Xeon– 3.3 GHz 12 Cache  2 Quad Core CPU  64 GB Ram  146 GB partición primaria para el OS (RAID1 2x146 15k RPM GB SAS) Dual 1GB Tarjeta de Red |

**Configuración Recomendada Servidor de Base de Datos:**

|  |
| --- |
| Características Hardware / Comentarios |
| Procesador Xeon– 2.4 GHz 16 MB Cache  4 Quad Core CPU  32 GB RAM  Partición 100 GB para OS  146 GB partición primaria para OS (RAID1 2x146 15k RPM GB SAS)  600 GB partición primaria (RAID5 3x300 15k RPM GB SAS)  Dual 1GB Tarjeta de Red |

**Configuración Recomendada Estación de Trabajo Usuario Final:**

|  |
| --- |
| Procesador compatible Intel  Memoria RAM 4 GB  Monitor a Color SVGA en adelante  **Máquina Virtual de Java versión 1.6**  **Internet Explorer versión 11 en adelante.**  **Google Chrome v. 48.x+**  Nota: la máquina virtual de Java y la versión del explorador de internet deben ser ambas de 64 bits o de 32 bits. |

**Turbinas de Base de Datos Soportadas:**

|  |
| --- |
| Oracle 9.x en adelante hasta 12.c |
| Microsoft Sql Server 2005, 2008, 2012 |
| Postgres SQL 8.3 hasta 9.3 |

**Servidores Web Soportados:**

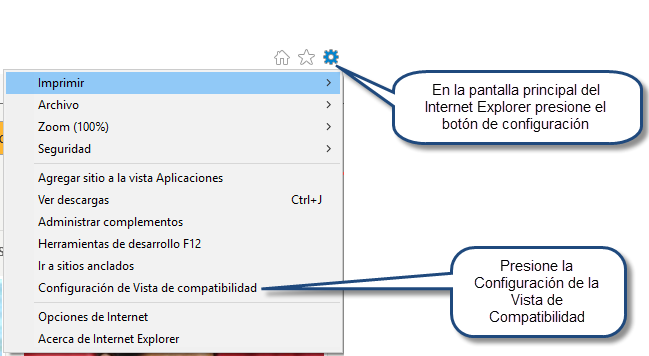
|  |
| --- |
| Apache Tomcat 7 y 8 |
| IBM Websphere 8.5 |
| JBoss EAP versiones 6+ y JBoss AS 7 + |

**Ancho de Banda Requerido**: por cada usuario Strategos consume 200 kb por usuario de ancho de banda.

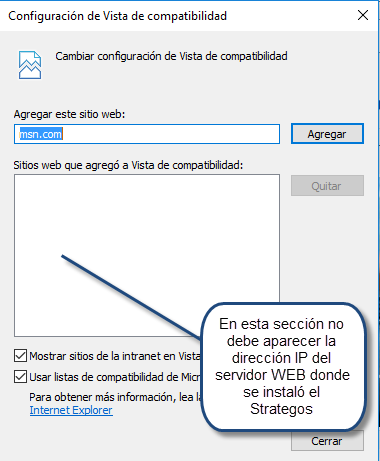
### Configuración del Internet Explorer

A continuación se presenta cómo debe estar configurado el explorador de Internet Explorer para que el Strategos® opere en forma adecuada.

El sistema opera con la versión del Internet Explorer 11, versiones anteriores al IE 11 no son compatibles con esta versión del Strategos, ya que no permiten correr códigos en JavaScript, el graficador y el generador de árboles de Dupont están escritos en Javascript.



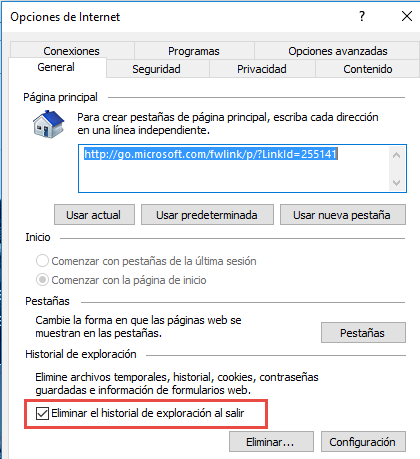
Verifique en la ventana siguiente que la dirección IP del servidor web donde está instalado el Strategos **NO ESTE PRESENTE**. Las versiones del Strategos liberadas a partir del 30 de abril de 2016 no utilizan la vista de compatibilidad, ya que sus pantallas están desarrolladas con HTML5; por lo que si deja activa la vista de compatibilidad el Strategos no funcionará correctamente.

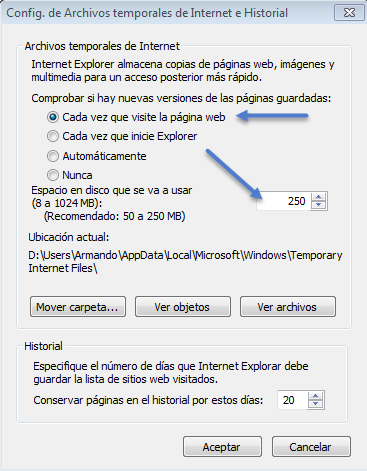


**Configuración Opciones de Internet (Internet Options):**

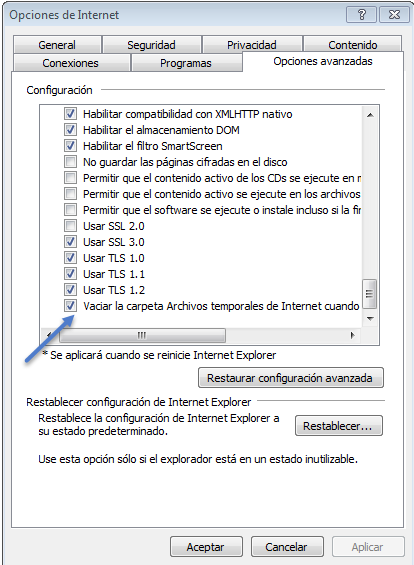
**General…**

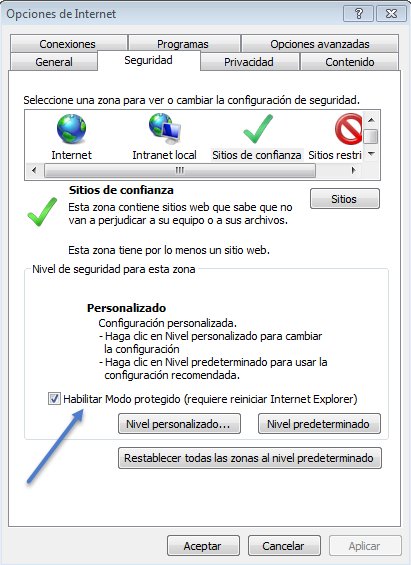
Esta configuración es la configuración normal que toda aplicación web debe tener, no es un estándar de Strategos®, sino de cualquier aplicación Web.

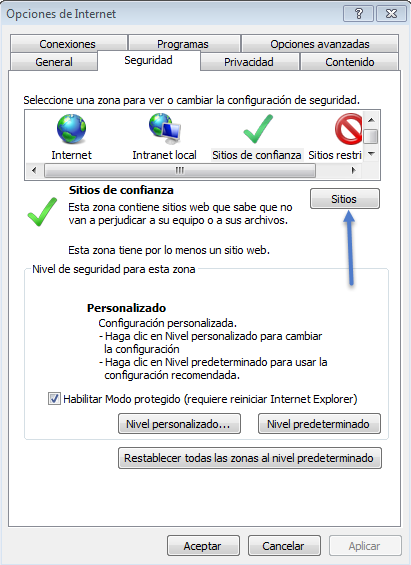




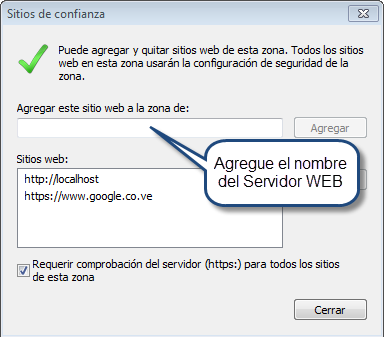
**Avanzada… (Seguridad)**

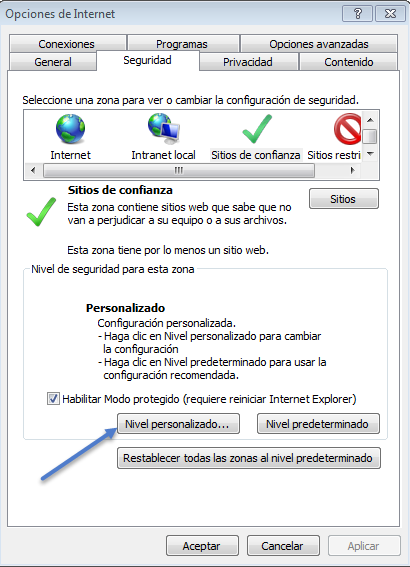


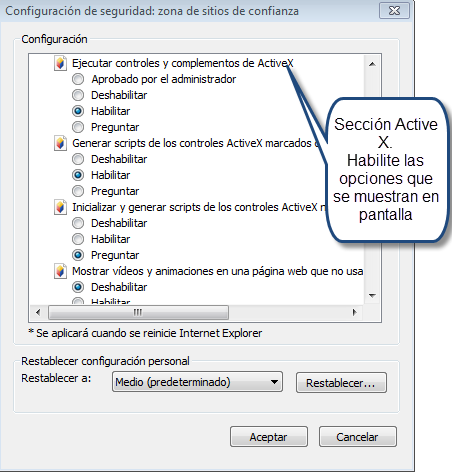




Presione el botón SITIOS

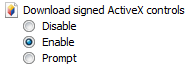


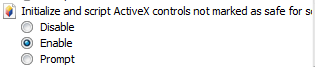


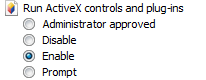


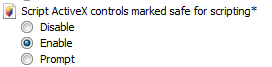
Presione Aceptar para terminar.

Si su versión del IE está en idioma inglés, estas son las opciones que deben ser activadas:





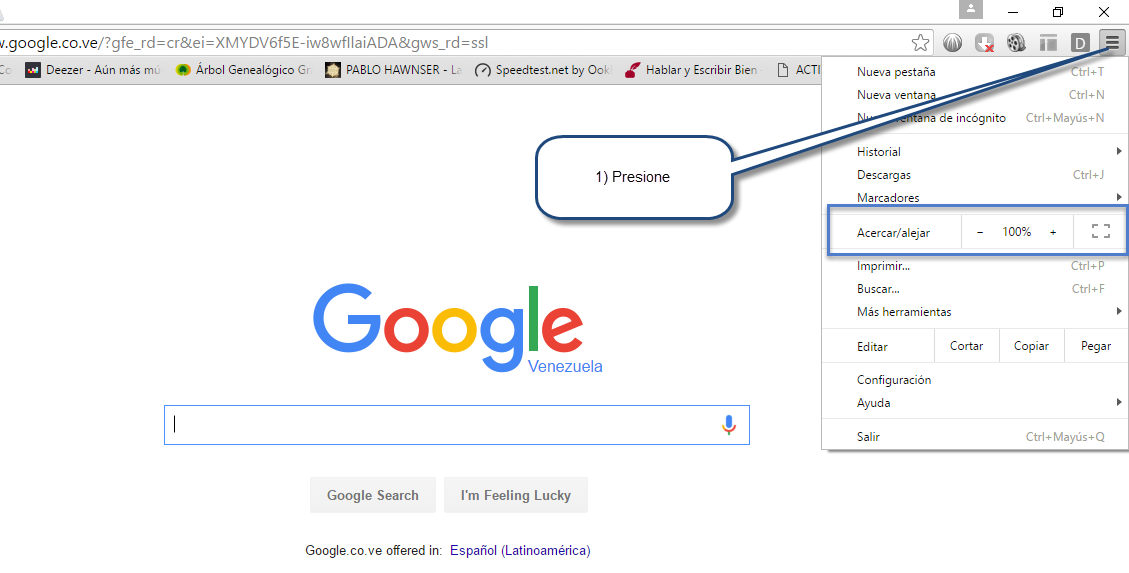




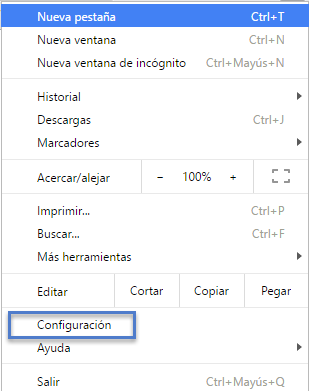
## Configuración para Google Chrome

A continuación se presenta cómo debe estar configurado el explorador de Internet Google Chrome para que el Strategos® opere en forma adecuada.

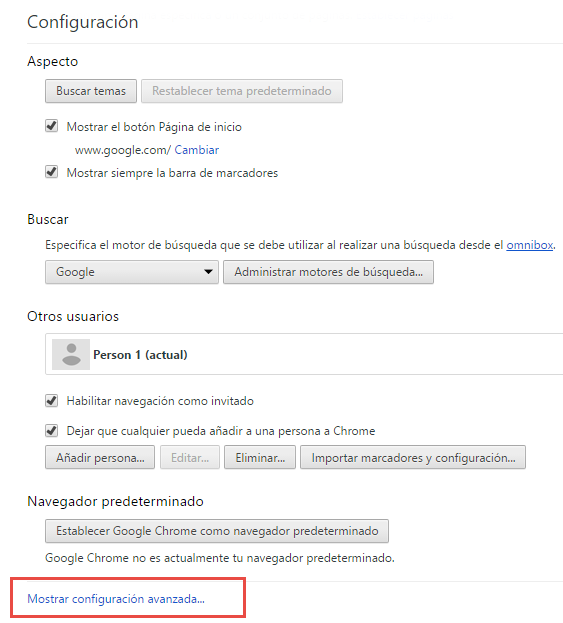
1. La resolución del explorador debe estar en 100%, mientras se use el Strategos.



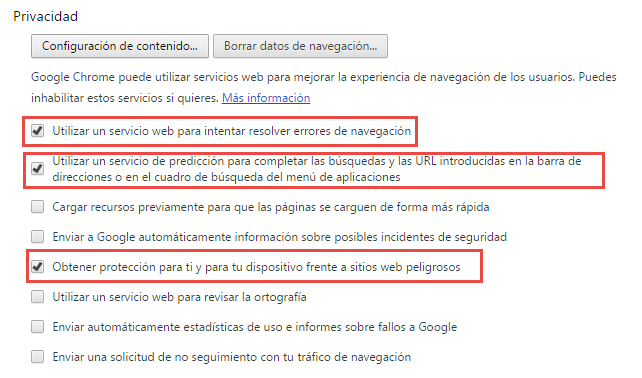
1. En el menú de configuración anterior seleccione la opción CONFIGURACIÓN

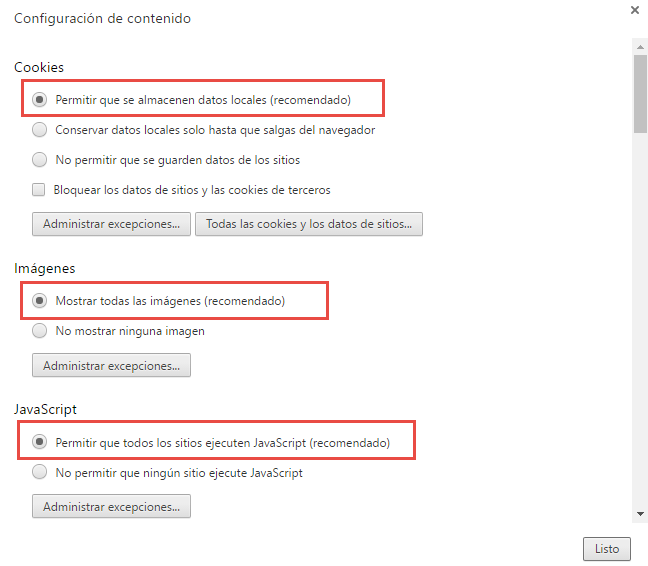


1. Seleccione la opción de Ahora en la configuración y configuración avanzada hay que tener lo siguiente

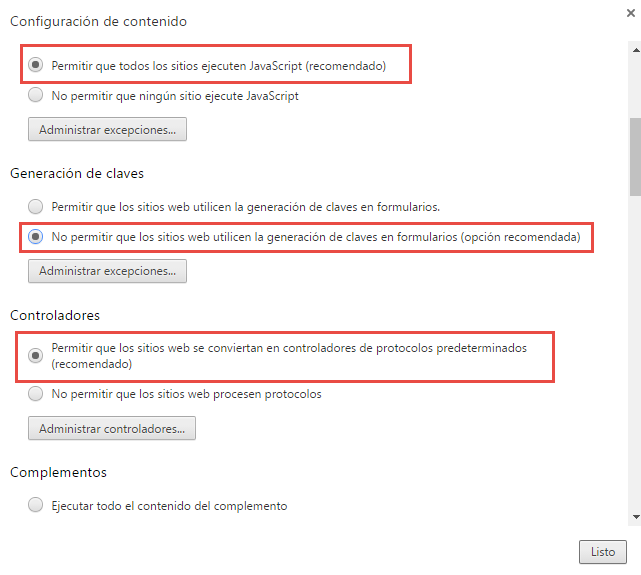


1. Seleccione la opción PRIVACIDAD y seleccione las opciones que se le presentan resaltadas en color rojo en la pantalla:

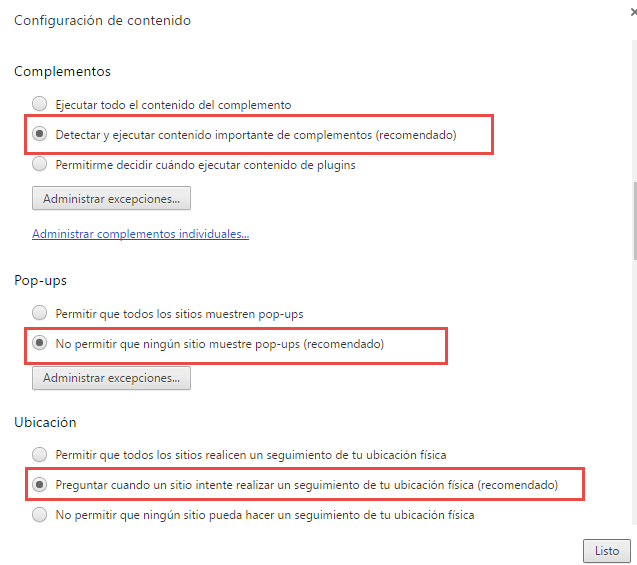


1. En la misma ventana anterior, presione el botón seleccione las opciones siguientes:  
   

Más abajo en la misma pantalla:



Más abajo en la misma pantalla:

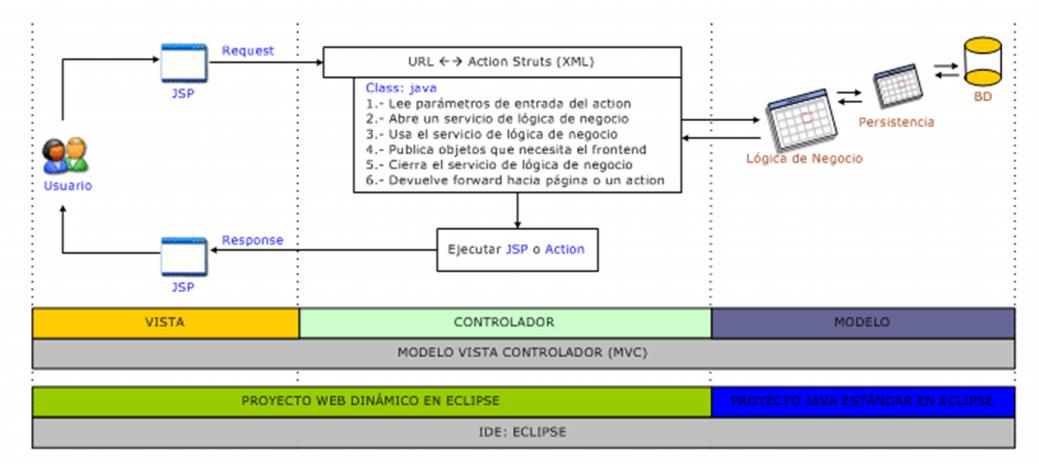


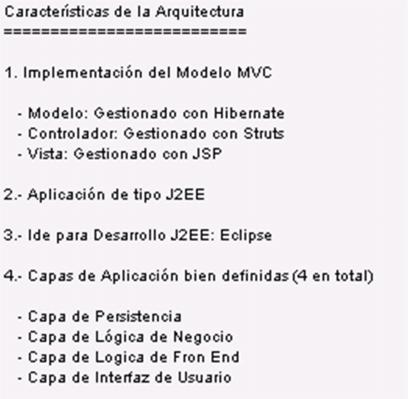
Presione el botón  para grabar los cambios. Salga del Google Chrome y vuelva a entrar para que tome la configuración que ha definido.

Fin del proceso.

### Arquitectura General del Strategos Advance

#### Diagrama de Arquitectura





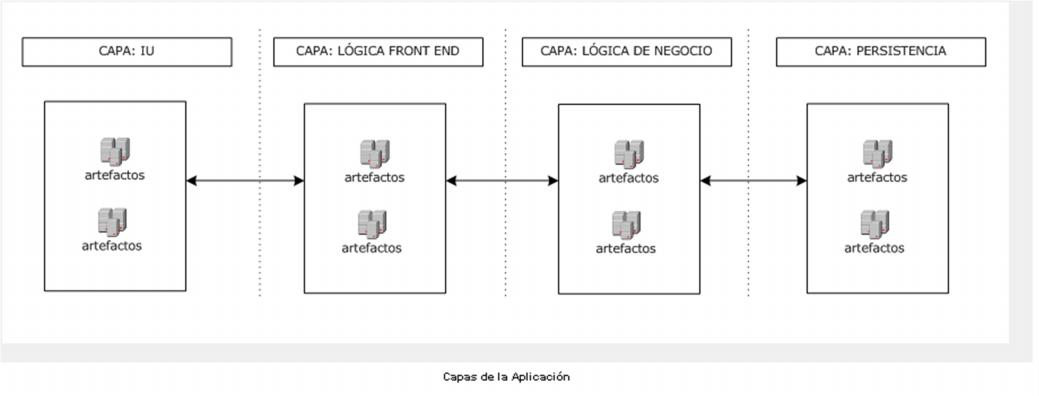
*Explicando el diagrama de arquitectura de la solución*: hay diferentes implementaciones de **mvc**, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con **la interfaz de usuario** a través de una página **jsp**.
2. El **controlador** recibe la notificación de la acción solicitada por el usuario. El **controlador** gestiona el evento que llega, frecuentemente a través de un gestor de eventos.
3. El **controlador** accede al **modelo**, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.
4. El **controlador** delega a los objetos de la vista la tarea de desplegar la **interfaz de usuario**. La vista obtiene sus datos del **modelo** para generar la interfaz apropiada para el usuario donde se refleja los cambios en el **modelo**. El **modelo** no debe tener conocimiento directo sobre la **vista**. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta interacción entre el **modelo** y la **vista**, permitiendo al **modelo** notificar a los interesados de cualquier cambio. Un objeto **vista** puede registrarse con el **modelo** y esperar a los cambios, pero aun así el **modelo** en sí mismo sigue sin saber nada de la **vista**. El **controlador** no pasa objetos de dominio (el modelo) a la **vista** aunque puede dar la orden a la **vista** para que se actualice.

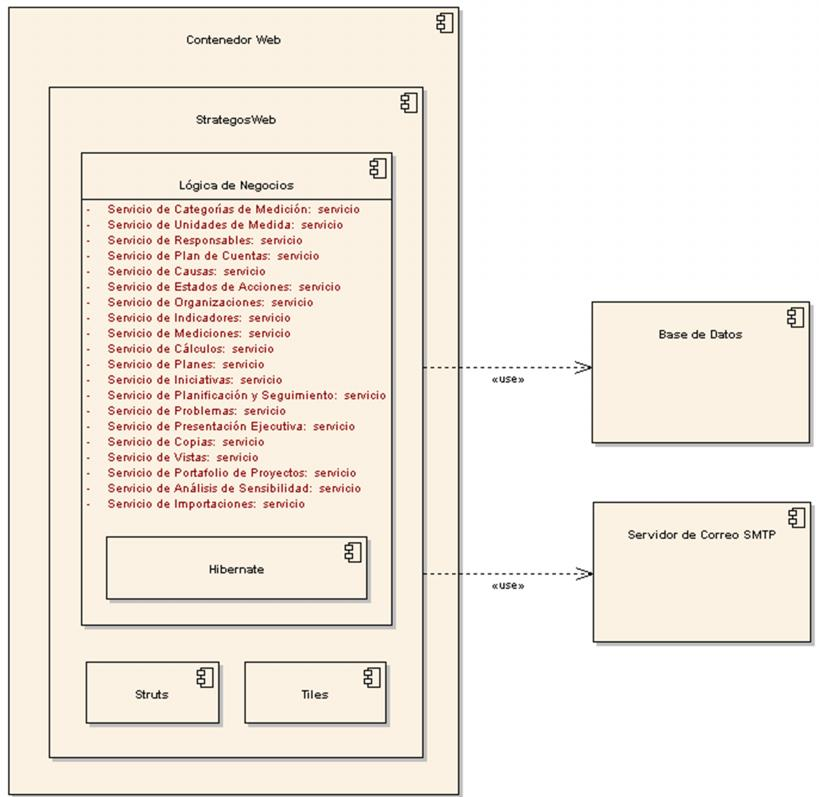
**NOTA**: en algunas implementaciones la **vista** no tiene acceso directo al modelo, dejando que el controlador envíe los datos del **modelo** a la **vista**.

1. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Diagrama Lógico de las Capas de la Aplicación



#### Modelo de Componentes



#### Estándares de Nomenclatura de Código

**Nombre de Paquete Base**

Los nombres de los paquetes siempre se escriben en letras minúsculas.

Patrón:

El nombre del paquete base de un componente web se compone de tres (3) partes:

1.- Prefijo ***com.visiongc.app***.

2.- Nombre del sistema, en minúsculas.

3.- Sufijo ***web***.

Ejemplos: com.visiongc.app.strategos.web

**Nombre de Paquete de Struts**

Patrón:

El nombre del paquete struts de un componente web se compone de dos (2) partes:

1.- Paquete base.

2.- Sufijo ***struts***.

Ejemplos: com.visiongc.app.strategos.web.struts

**Nombre de Paquete de Actions de Struts**

Patrón:

El nombre de un paquete de Actions de Struts de un componente web se compone de tres

(3) partes:

1.- Paquete struts.

2.- Nombre de módulo del sistema, si aplica.

3.- Sufijo ***actions***.

Ejemplos: com.visiongc.app.strategos.web.struts.unidadesmedida.actions com.visiongc.app.strategos.web.struts.categoriasmedicion.actions com.visiongc.app.strategos.web.struts.causas.actions

**Nombre de Paquete de FormBeans de Struts**

Patrón:

El nombre de un paquete de FormBeans de Struts de un componente web se compone de tres (3) partes:

1.- Paquete struts.

2.- Nombre de módulo del sistema, si aplica.

3.- Sufijo ***forms***.

Ejemplos: com.visiongc.app.strategos.web.struts.unidadesmedida.forms com.visiongc.app.strategos.web.struts.categoriasmedicion.forms com.visiongc.app.strategos.web.struts.causas.forms

**Nombre de Clase de Configuración de Permisos (MapStrutsActionRight)**

Patrón:

El nombre de una clase de Configuración de Permisos asociados a Actions de Struts se compone de dos (2) partes:

1.- Nombre del Sistema, comenzando con mayuscula.

2.- Sufijo ***MapStrutsActionsRight***.

Ejemplos: StrategosMapStrutsActionRight

**Nombre de ActionRight (Clase Java de Permisología)**

Patrón:

El nombre de un ActionRight se obtiene del Mapeo de Action de Struts sobre el que se debe verificar el permiso. Se compone de palabras en ***minúsculas*** separadas por ***puntos (.)***, donde los puntos reemplazan las ***barras (/)*** que se encuentran en los nombres de los Mapeos de los Actions de Struts.

Ejemplos:

Action de Struts

<action path="/unidadesmedida/crearUnidadMedida" >

ActionRight

actionRight = **new** StrutsActionRight("unidadesmedida.crearunidadmedida", **false**, **false**, "UNIDAD");

mapa.put("unidadesmedida.crearunidadmedida", actionRight);

**Nombre de Action de Struts (Clases Java)**

Patrón:

El nombre de un Action de Struts se compone de tres (3) partes:

1.- Verbo en infinitivo que indica la acción que se ejecuta.

2.- Nombre de un Objeto de Sistema o de otro tipo en notación Pascal en plural o singular, dependiendo del caso, sobre el que se ejecuta la acción. IMPORTANTE: Se debe tener cuidado con el uso de los nombres en PLURAL o SINGULAR. Siempre verificar contra el documento de diseño.

3.- Sufijo ***Action***.

Ejemplos: GestionarUnidadesMedidaAction

EditarUnidadMedidaAction

GuardarUnidadMedidaAction

EliminarUnidadMedidaAction

CalcularIndicadoresAction CalcularIndicadorAction

**Nombre de FormBean de Struts (Clases Java)**

Patrón:

El nombre de un FormBean de Struts se obtiene del nombre del Action de Struts que lo utiliza. El nombre de un Action de Struts se compone de dos (2) partes:

1.- Nombre del Action de Struts que utiliza el FormBean ***sin el sufijo Action***.

2.- Sufijo ***Form***.

Ejemplos: GestionarUnidadesMedidaForm

EditarUnidadMedida

Form CalcularIndicadores

**Nombre de Archivo de Recursos de Lenguaje**

Patrón:

El nombre del Archivo de Recursos de Lenguaje de un componente web se compone de cuatro (4) partes:

1.- Palabra ***MessageResources***.

2.- Nombre del Sistema en notación Pascal.

3.- Palabra ***Web***.

4.- Extensión ***properties***.

Ejemplo: MessageResourcesStrategosWeb.properties MessageResourcesVernierWeb.properties MessageResourcesRadarWeb.properties

**Clave de Recurso de Lenguaje (Archivos de Recursos de Lenguaje)**

Los recursos de lenguaje se pueden clasificar en tipos, de acuerdo a en que componentes se utilizan o que información contienen, a saber:

1.- Los que contienen información del sistema o componente web.

2.- Los que se utilizan en Actions de Struts (Clases Java).

3.- Los que se utilizan en páginas JSP.

4.- Los que se utilizan para darle nombres a botones que se utilizan en páginas JSP.

5.- Los que se utilizan para darle nombres a las opciones de menú en páginas JSP.

6.- Para guardar patrones de formato (fechas, números, etc.).

7.- Los que se utilizan para darles nombre a puntos de navegación en la barra de navegación.

Patrón:

Las claves siempre se escriben con múltiples palabras en minúsculas, separando las palabras con puntos cuando sea necesario. Existe un patrón por tipo de recurso de lenguaje:

1.- De información de sistema o componente web: Se componen de dos (2) partes:

1.1.- Palabra ***aplicación***.

1.2.- Palabra o palabras que identifiquen la información que contiene el recurso de lenguaje.

Ejemplos: aplicacion.nombre=Strategos apliccion.autor=Vision Grupo Consultores C.A. aplicacion.copyright=Copyright &copy; aplicacion.ano=2008 aplicacion.tituloversion=Versión aplicacion.version=beta aplicacion.version.fecha=14/03/2013

2.- De Actions de Struts: Se componen de tres (3) partes:

2.1.- Palabra ***action***.

2.2.- Nombre del action donde se utiliza el recurso de lenguaje ***sin el sufijo action***.

2.3.- Palabra o palabras que identifiquen el mensaje que contiene el recurso de lenguaje.

Ejemplos:

action.eliminarregistro.relacionados=El objeto no puede ser eliminado porque tiene objetos relacionados

action.eliminarregistro.bloqueado=El objeto está siendo utilizado y no se puede eliminar action.eliminarregistro.exito=El objeto ha sido eliminado exitosamente action.eliminarregistro.noencontrado=El objeto fue eliminado por otro usuario

3.- De páginas JSP: Se componen de tres (3) partes:

3.1.- Palabra ***jsp***.

3.2.- Nombre de la página JSP donde se utiliza el recurso de lenguaje.

3.3.- Palabra o palabras que identifiquen el mensaje que contiene el recurso de lenguaje.

Ejemplos:

jsp.editarunidadmedida.titulopagina.nuevo=Nueva Unidad de Medida jsp.editarunidadmedida.titulopagina.modificar=Modificar Unidad de Medida jsp.editarunidadmedida.ficha.nombre=Nombre

4.- De botones: Se componen de dos (2) partes:

4.1.- Palabra ***boton***.

4.2.- Palabra o palabras que identifiquen la acción que ejecuta el botón.

Ejemplos:

boton.regresar=Regresar boton.limpiar=Limpiar boton.buscar=Buscar

boton.cancelar=Cancelar

boton.aceptar=Aceptar

5.- De menú: Se componen de dos (2) partes:

5.1.- Palabra ***menu***.

5.2.- Palabra o palabras que identifiquen la acción que ejecuta la opción de menú.

Ejemplos:

menu.separador=&nbsp; menu.reporte=Reporte

menu.edicion=Edición menu.edicion.nuevo=Nuevo menu.edicion.modificar=Modificar

menu.edicion.eliminar=Eliminar menu.ver=Ver

menu.ver.unidadesmedida=Unidades de Medida

menu.ver.categoriasmedicion=Categorías de Medición

6.- De patrones de formato: Se componen de dos (2) partes:

5.1.- Palabra ***formato***.

5.2.- Palabra o palabras que identifiquen el patrón de formato que contiene el recurso de lenguaje.

Ejemplos: formato.hora.corta=HH:mm:ss

formato.hora.larga=HH:mm:ss

formato.fecha.corta=dd-MM-yyyy

formato.fecha.larga=dd-MM-yyyy

formato.timestamp=dd-MM-yyyy HH:mm:ss

formato.numero.condecimal=###,##0.00

7.- De puntos de navegación: El nombre de un punto de navegación se obtiene del Mapeo de Action de Struts que se debe agregar como punto de navegación. Se compone de palabras en ***minúsculas*** separadas por ***puntos (.)***, donde los puntos reemplazan las ***barras (/)*** que se encuentran en los nombres de los Actions de Struts.

Ejemplo:

Action de Struts

<action path="/unidadesmedida/gestionarUnidadesMedida" > Clave de Recurso de Lenguaje unidadesmedida.gestionarunidadesmedida=Gestionar Unidades de Medida

**Nombre de Página JSP**

Patrón:

El nombre de una página JSP se compone de dos (2) partes:

1.- Verbo en infinitivo que indica la acción que se ha ejecutado y que ha llamado a la página JSP para presentación de resultado, comenzando con minúscula.

2.- Nombre de un Objeto de Sistema o de otro tipo en notación Pascal en plural o singular, dependiendo del caso, sobre el que se ejecuta la acción. IMPORTANTE: Se debe tener cuidado con el uso de los nombres en PLURAL o SINGULAR. Siempre verificar contra el documento de diseño.

Ejemplos:

gestionarUnidadesMedida.jsp

editarUnidadMedida.jsp

Caso Especial

Existe una página JSP que contiene funciones javascript de un módulo del sistema. Esta página sigue otro patrón para su nombre, que se compone de dos (2) partes:

1.- Nombre del módulo respectivo, comenzando con minúscula.

2.-Sufijo ***Js***.

Ejemplos:

unidadesMedidaJs.jsp

categoriasMedicionJs.jsp

causasJs.jsp

**Nombre de Archivo de Configuración de Struts**

Patrón:

El nombre de un Archivo de Configuración de Struts se compone de seis (6) partes separadas por guiones:

1.-Palabra ***struts***.

2.- Nombre del Sistema en minúsculas.

3.-Palabra ***Web***.

4.- Nombre del módulo en minúsculas.

5.- Palabra ***config***.

6.-Extensión ***xml***.

Ejemplos:

struts-strategosWeb-unidadesmedida-config.xml struts-strategosWeb-categoriasmedicion-config.xml struts-strategosWeb-estadosacciones-config.xml

**Mapeo de FormBean de Struts (Archivos de configuración de Struts)**

Patrón:

El nombre de un FormBean de Struts se compone de dos (2) partes:

1.- Nombre del FormBean en notación Pascal, con la primera letra en minúscula. El nombre generalmente se obtiene del nombre del FormBean de Struts (Clase Java) que instancia el FormBean. En el caso de utilizar FormBeans genéricos se utiliza el nombre del Action de Struts que utilizará el FormBean.

2.-Palabra ***Form***.

Ejemplo:

<form-bean name="editarUnidadMedidaForm" type="com.visiongc.app.strategos.web.struts.unidadesmedida.forms.EditarUnidadMedidaFor m" />

<form-bean name="gestionarUnidadesMedidaForm" type="com.visiongc.app.strategos.web.struts.unidadesmedida.forms.GestionarUnidadesMedi daForm" />

<form-bean name="seleccionarUnidadesMedidaForm" type="com.visiongc.framework.web.struts.forms.SelectorListaForm" />

**Mapeo de Action de Struts (Archivos de configuración de struts)**

Patrón:

El nombre de un action se compone de dos partes:

1.- Nombre del módulo en notación Pascal.

2.- Nombre del Caso de Uso en notación Pascal que implementa el Action de Struts seguido del sufijo Action. Los nombres de los Casos de Uso se componen a su vez de dos partes:

2.1.- Verbo en infinitivo que indica la acción que se ejecuta.

2.2.- Nombre de un Objeto de Sistema o de otro tipo en notación Pascal sobre el que se ejecuta la acción.

Ejemplos:

/unidadesmedida/gestionarUnidadesMedida

/unidadesmedida/crearUnidadMedida

/unidadesmedida/modificarUnidadMedida

/unidadesmedida/guardarUnidadMedida

/unidadesmedida/eliminarUnidadMedida

/unidadesmedida/generarReporteUnidadesMedida

/unidadesmedida/seleccionarUnidadesMedida

#### Componente de Lógica de Negocio

**Nombre de Paquete Base**

Los nombres de los paquetes siempre se escriben en letras minúsculas.

Patrón:

El nombre del paquete base de un componente de lógica de negocios se compone de dos (2)

partes:

1.-Prefijo ***com.visiongc.app***.

2.- Nombre del sistema, en minúsculas.

Ejemplos: com.visiongc.app.strategos

**Nombre de Paquete de Módulo**

Patrón:

El nombre de un paquete de módulo de un componente de lógica de negocios se compone de dos (2) partes:

1.- Paquete base.

2.- Nombre del módulo, en minúsculas.

Ejemplos: com.visiongc.app.strategos.unidadesmedida com.visiongc.app.strategos.causas com.visiongc.app.strategos.organizaciones

**Nombre de Paquete de Implementación de Módulo**

Patrón:

El nombre de un paquete de implementación de un módulo de un componente de lógica de negocios se compone de dos (2) partes:

1.- Paquete del módulo.

2.- Sufijo ***impl***.

Ejemplos: com.visiongc.app.strategos.unidadesmedida.impl com.visiongc.app.strategos.causas.impl com.visiongc.app.strategos.organizaciones.impl

**Nombre de Paquete de Modelo de Módulo**

Patrón:

El nombre de un paquete de modelo de un módulo de un componente de lógica de negocios se compone de dos (2) partes:

1.- Paquete del módulo.

2.- Sufijo ***model***.

Ejemplos: com.visiongc.app.strategos.unidadesmedida.model com.visiongc.app.strategos.causas.model com.visiongc.app.strategos.organizaciones.model

**Nombre de Paquete Utilitario de Modelo de Módulo**

Patrón:

El nombre de un paquete utilitario de modelo de un módulo de un componente de lógica de negocios se compone de dos (2) partes:

1.- Paquete del módulo.

2.- Sufijo ***model.util***.

Ejemplos: com.visiongc.app.strategos.unidadesmedida.model.util com.visiongc.app.strategos.causas.model.util com.visiongc.app.strategos.organizaciones.model.util

**Nombre de Paquete de Persistencia de Módulo**

Patrón:

El nombre de un paquete de persistencia de un módulo de un componente de lógica de negocios se compone de dos (2) partes:

1.- Paquete del módulo.

2.- Sufijo ***persistence***.

Ejemplos: com.visiongc.app.strategos.unidadesmedida.persistence com.visiongc.app.strategos.causas.persistence com.visiongc.app.strategos.organizaciones.persistence

**Nombre de Paquete de Persistencia de Hibernate de Módulo**

Patrón:

El nombre de un paquete de persistencia de hibernate de un módulo de un componente de lógica de negocios se compone de dos (2) partes:

1.- Paquete del módulo.

2.- Sufijo ***persistence.hibernate***.

Ejemplos: com.visiongc.app.strategos.unidadesmedida.persistence com.visiongc.app.strategos.causas.persistence com.visiongc.app.strategos.organizaciones.persistence

**Nombre de Paquete Utilitario de Módulo**

Patrón:

El nombre de un paquete utilitario de un módulo de un componente de lógica de negocios se compone de dos (2) partes:

1.- Paquete del módulo.

2.- Sufijo ***util***.

Ejemplos:

com.visiongc.app.strategos.unidadesmedida.model.util com.visiongc.app.strategos.causas.model.util com.visiongc.app.strategos.organizaciones.model.util

**Nombre de Archivo de Recursos de Lenguaje**

Patrón:

El nombre del Archivo de Recursos de Lenguaje de un componente de lógica de negocio se compone de tres (3) partes:

1.- Palabra ***MessageResources***.

2.- Nombre del Sistema en notación Pascal.

3.- Extensión ***properties***.

Ejemplo: MessageResourcesStrategos.properties

**Nombre de Archivo de Parámetros de Configuración**

Patrón:

El nombre del Archivo de Parámetros de Configuración de un componente de lógica de negocio se compone de dos (2) partes:

1.- Nombre del Sistema o componente en notación Pascal.

2.- Extensión ***properties***.

Ejemplo: Strategos.properties

**Nombre de Clase de Manejo de Parámetros de Configuración**

Patrón:

El nombre de la Clase de Manejo de Parámetros de Configuración de un componente de lógica de negocio se compone de dos (2) partes:

1.- Nombre del Sistema o componente en notación Pascal.

2.- Sufijo ***Configuration***.

Ejemplo: StrategosConfiguration

**Nombre de Interface de Servicio**

Patrón:

El nombre de una Interface de Servicio de un componente de lógica de negocio se compone de tres (3) partes:

1.- Nombre del Sistema o componente en notación Pascal.

2.- Nombre del Módulo en notación Pascal, si aplica.

3.- Sufijo ***Service***.

Ejemplo: StrategosService

StrategosUnidadesMedidaService StrategosCausasService

**Nombre de Implementación de Servicio**

Patrón:

El nombre de una Implementación de Servicio de un componente de lógica de negocio se compone de dos (2) partes:

1.- Nombre de la interface que implementa.

2.- Sufijo ***Impl***.

Ejemplo: StrategosServiceImpl StrategosUnidadesMedidaServiceImpl StrategosCausasServiceImpl

**Nombre de Clase de Modelo**

Patrón:

El nombre de una Clase de Modelo de un componente de lógica de negocio se compone de una (1) parte:

Nombre del Objeto Conceptual de Sistema o componente, en singular y en notación Pascal.

Ejemplo: UnidadMedida Causa Organización

**Nombre de Archivo de Mapeo de Clase de Modelo de Hibernate**

Patrón:

El nombre de un Archivo de Mapeo de Clase de Modelo de Hibernate de un componente de lógica de negocio se compone de dos (2) partes:

1.- Nombre del Objeto Conceptual de Sistema o componente, en singular y en notación Pascal.

2.- Extensión ***hbm.xml***.

Ejemplo: UnidadMedida.hbm.xml

Causa.hbm.xml Organizción.hbm.xml

**Nombre de Interface de Servicio de Persistencia**

Patrón:

El nombre de una Interface de Servicio de Persistencia de un componente de lógica de negocio se compone de tres (3) partes:

1.- Nombre del Sistema o componente en notación Pascal.

2.- Nombre del Módulo en notación Pascal, si aplica.

3.- Sufijo ***PersistenceSession***.

Ejemplo: StrategosPersistenceSession StrategosUnidadesMedidaPersistenceSession StrategosCausasPersistenceSession

**Nombre de Implementación de Servicio Persistencia con Hibernate**

Patrón:

El nombre de una Implementación de Servicio de Persistencia con Hibernate de un componente de lógica de negocio se compone de tres (3) partes:

1.- Nombre del Sistema o componente en notación Pascal.

2.- Nombre del Módulo en notación Pascal, si aplica.

3.- Sufijo ***HibernateSession***.

Ejemplo: StrategosHibernateSession StrategosUnidadesMedidaHibernateSession StrategosCausasHibernateSession

Notación Pascal: Nombres compuestos de múltiples palabras juntas, iniciando cada palabra con letra mayúscula.

# CAPITULO III

## INSTALACIÓN Y CONFIGURACIÓN DEL AMBIENTE DE DESARROLLO EN PLATAFORMA WINDOWS

El siguiente documento tiene como propósito fundamental, explicar los pasos que se deben seguir para llevar a cabo la instalación y configuración del ambiente de desarrollo en la organización**.**

La información que se expresa acá es de vital importancia, puesto que estandariza la configuración del ambiente de desarrollo, así como también todos los elementos de apoyo al mismo, tal como; ides, estructura de directorios, variables, bases de datos, librerías, entre otros.

Convenciones Utilizadas:

Con el objeto de mejorar el rendimiento en la compresión de ideas en esta documentación, se crearon algunas convenciones que pondrán alerta al lector con respeto a lo que ocurre en el transcurso de la implementación. Tales convenios son los siguientes:

**NOTA**: Los recuadros sombreados con la indicación de **NOTA** muestran información de importancia y que el lector deberá recordar y que es sumamente relevante dentro del contexto que se introduce.

*El estilo cursiva se utiliza para comentarios al margen del tema que está siendo tratado.*

Los estilo del texto:

* + - Al introducirlas, **destacamos** en negrita las palabras importantes.
    - Mostramos las combinaciones de teclado así: *Control+A* o *Shift+Supr*
    - Mostramos los nombres de los archivos o carpetas así: “nombreArchivo” o “nombreCarpeta”
    - Mostramos los códigos y líneas de comando en el texto así: código() o línea de comando
    - Se debe respetar el uso de altas y bajas al momento de leer la información.
    - Los textos que se encuentran entre corchetes son explicativos de la acción que debe tomar el lector. Ejemplos:

[vacio]: Indica que debe mantener el valor del elemento completamente vacio.

[clave]: Indica que se debe introducir una clave a conveniencia del lector.

En este recuadro se escribirán los comandos y códigos que deben ser tipeados de forma idéntica bien sea en bloc de notas, cualquier editor de textos o consola.

Los texto rojos que se encuentran entre numerales no deben ser escrito en el código, estos solo son explicativos

# Comentario de ejemplo #

<!—Comentario de ejemplo -->

.estilo {

color:black;

font-size:12px;

otro:12xx

}

### Instalación del Ambiente de Desarrollo

#### Creación de la Estructura de Directorios

Antes de iniciar la instalación es necesario que se cree la siguiente estructura de directorios.

**NOTA**: para este ejemplo se asume que toda la instalación se hace en el disco **F:/** , pero esta particularidad no es obligatoria. Esto quiere decir que puede ser en el disco **C:/** o cualquier otro que tenga espacio suficiente para albergar la instalación.

* El espacio mínimo requerido es de 550 MB.
* El espacio recomendado es de 1 GB.

Los productos que van a ser instalados son los siguientes:

* + - JDK.
    - Apache Tomcat y sus librerías necesarias.
    - PostgreSQL o SQlServer y sus librerías necesarias.
    - Eclipse y los plugins necesarios.

Es importante recalcar que este manual llevará a cabo el proceso de instalación en el mismo orden en que se presentaron los elementos de la lista anterior.

**NOTA**: se asume que el lector tiene conocimientos básicos sobre los productos que se van a instalar, por tal razón no se hace una definición de estos.

Primeramente se debe crear los siguientes directorios:

* + - F:/ambienteDesarrollo
    - F:/ambienteDesarrollo/java
    - F:/ambienteDesarrollo/ides
    - F:/ambienteDesarrollo/contenedorServlet
    - F:/ambienteDesarrollo/servidorAplicaciones
    - F:/ambienteDesarrollo/baseDatos
    - F:/ambienteDesarrollo/ides/eclipse
    - F:/ambienteDesarrollo/proyectos/eclipse

#### Instalación y configuración del Jdk

Se asume que ya se ha realizado la descarga del producto desde un sitio web confiable y autorizado. La instalación del producto puede variar dependiendo de la versión descargada, pero en la mayoría de los casos son cambios poco relevantes y no afectan la vigencia de este manual.

Se asume que el archivo se encuentra en el siguiente directorio:

F:/paqueteInstalacion/jdk-6u2-windows-i586-p.exe

**🡪 Pasos de la instalación y configuración**:

* + - Ejecutar el archivo “jdk-6u2-windows-i586-p.exe”.
    - Hacer la instalación en el directorio: F:/ambienteDesarrollo/java/
    - Una vez que finalice la instalación se debe registrar la variable de entorno con el siguiente procedimiento:
      * Ir a la siguiente ubicación Mi PC 🡪 Propiedades 🡪 Opciones Avanzadas 🡪 Variables de Entorno. *Ver figura n° 1*
      * Se debe crear una **Variable de Sistema** llamada JAVA\_HOME y su valor debe ser la ruta donde se instaló el **jdk**, es decir; F:/ambienteDesarrollo/java/
    - Una vez que se haya realizado esta operación aceptamos los cambios y queda instalado y configurado **jdk**.



#### Instalación y configuración de Apache Tomcat

Se asume que ya se ha realizado la descarga del producto desde un sitio web confiable y autorizado. La instalación del producto puede variar dependiendo de la versión descargada, pero en la mayoría de los casos son cambios poco relevantes y no afectan la vigencia de este manual.

**Figura n° 1**: Mi PC 🡪 Propiedades

Se asume que el archivo se encuentra en el siguiente directorio:

F:/paqueteInstalacion/apache-tomcat-6.0.14.exe

**🡪 Pasos de la instalación y configuración**:

* + - Ejecutar el archivo “apache-tomcat-6.0.14.exe”.
    - Hacer la instalación en el directorio:

F:/ambienteDesarrollo/contenedorServlet/Tomcat.6.0

* + - Establecer los siguientes parámetros durante la instalación
      * **Tipo de instalación**: Full
      * **Connector Port**: 8080
      * **User Name**: admin
      * **Password**: 12345
      * **Ruta del J2SE o JRE**: F:/ambienteDesarrollo/java/
    - Una vez que se haya terminado este procedimiento queda instalado y configurado **Apache Tomcat**.

**NOTA**: este instalador hace el registro del contenedor de servlets (**Apache Tomcat**) en los **Servicios** de **Microsoft Windows XP**. De igual manera, queda listo y configurado para iniciar de forma automática cada vez que se enciende el equipo.

#### Instalación y configuración de Postgresql

Se asume que ya se ha realizado la descarga del producto desde un sitio web confiable y autorizado. La instalación del producto puede variar dependiendo de la versión descargada, pero en la mayoría de los casos son cambios poco relevantes y no afectan la vigencia de este manual.

Se asume que los archivos se encuentran en el siguiente directorio:

F:/paqueteInstalacion/postgreSQL/postgresql-8.2.msi

F:/paqueteInstalacion/postgreSQL/postgresql-8.2-int.msi

Para este producto hay una particularidad. Una vez que se ha descargado el **zip** y se descomprime, se puede apreciar que hay dos archivos **.msi**. Ambos archivos son necesarios para la instalación.

**🡪 Pasos de la instalación y configuración**:

* + - Ejecutar el archivo “postgresql-8.2.msi”.
    - Hacer la instalación en el directorio:

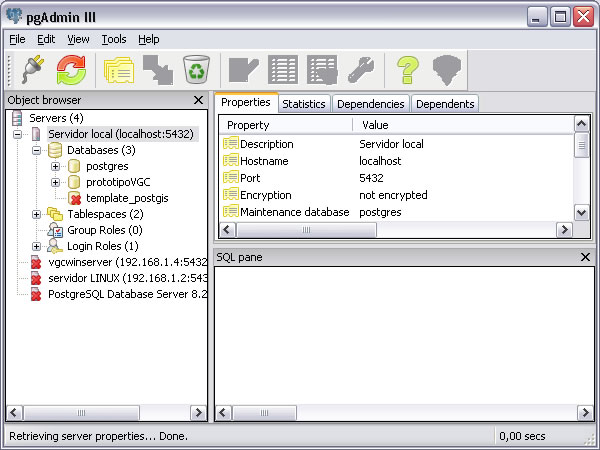
F:/ambienteDesarrollo/baseDatos/PostgreSQL/8.2/

* + - Establecer los siguientes parámetros durante la instalación
      * **Lenguaje**: English
      * **Instalar como servicio:** True
      * **Service name**: PostgreSQL Database Server 8.2
      * **Account name**: postgres
      * **Account domain:** [nombre equipo]
      * **Account password**: [vacio]
      * **Verify password**: [vacio]
      * **Initialize database cluster**: True
      * **Port number**: 5432
      * **Locale**: C
      * **Enconding**: SQL\_ASCII
      * **Superuser name**: sa
      * **Password**: vision
      * Marcar el check **Procedural languages PL/pgsql**
      * Marcar el check **Adminpacks**.
      * Marcar el check **Large Objects (lo)**.
    - Copiar el archivo “postgresql-8.3dev-601.jdbc4.jar” en el directorio:

F:/ambienteDesarrollo/contenedorServlet/Tomcat.6.0/common/lib/

* + - Una vez que se haya terminado este procedimiento queda instalado y configurado **PostgreSQL**.

**NOTA**: este instalador prepara y configura la aplicación **pgAdmin**. Este es un administrador visual que permite gestionar las bases de datos de **postgresql** desde una interfaz sencilla y amigable. *Ver figura n° 2*



**Figura n° 2**: administrador visual de PostgreSQL

#### Instalación y configuración de Eclipse

Se asume que ya se ha realizado la descarga del producto desde un sitio web confiable y autorizado. La instalación del producto puede variar dependiendo de la versión descargada, pero en la mayoría de los casos son cambios poco relevantes y no afectan la vigencia de este manual.

Se asume que el archivo se encuentre en el siguiente directorio:

F:/paqueteInstalacion/eclipse.zip

**NOTA**: para la configuración de **eclipse** en **Visión Grupo Consultores C.A.**, es necesario haber instalado con anterioridad **Microsoft Visual Source Safe**, puesto que es la herramienta que se utiliza en la organización para control de versiones. Se asume que el directorio donde se instaló **VSS** es:

C:/Archivos de programa/Microsoft Visual Studio/COMMON/VSS

**🡪 Pasos de la instalación y configuración:**

* + - Descomprimir el archivo “eclipse.zip” en el directorio:

F:/ambienteDesarrollo/ides/eclipse

* + - Copiar el directorio: F:/paqueteInstalacion/org.vssplugin\_1.6.1 y pegarlo en el directorio: F:/ambienteDesarrollo/ides/eclipse/plugins/
    - Copiar el archivo “srcsafe.ini” que se encuentra en el directorio: F:/paqueteInstalacion/ y reemplazar el original en la ruta del **VSS** que se especificó anteriormente.
    - Ejecutar (doble clic) el archivo “eclipse.exe” que se encuentra en el directorio:

F:/ambienteDesarrollo/ides/eclipse/

* + - Definir el espacio de trabajo (**Workspace**) en la siguiente ruta:

F:/ambienteDesarrollo/proyectos/eclipse

* + - Marcar el check **Utilizar este directorio como predeterminado**.
    - Establecer los siguientes parámetros en el menú de eclipse Window 🡪 Preferences
      * General 🡪 Web Browser: activar la opción **Usar browser externo**
      * Java🡪Editor🡪Installed JREs: agregar los datos del J2SDK
      * Server🡪Installed Runtimes: agregar el servidor **Apache Tomcat v7.0**
      * Team, establecer los siguientes parámetros:
        + Marcar el check **Show the file author in compare editors**.
        + Marcar el check **Reuse open compare editors when opening comparisons**.
        + La opción **Open the associated perspective when a synchronize operations complete** establecerla a **Always**.
      * En la pestaña **General** de Team🡪VSS, establecer los siguientes parámetros:
        + Marcar todas las opciones excepto **Ask for comment on check- outs**.
        + Establecer el tiempo de actualización automático a **45 minutes**.
    - Una vez que se haya terminado este procedimiento queda instalado y configurado **eclipse.**

### Plataforma con que fue Desarrollado el Strategos Advance

#### MVC

**Modelo Vista Controlador**, es un patrón de arquitectura de software que separa los datos de una aplicación, la **interfaz de usuario**, y la **lógica de control** en tres (3) componentes distintos. El patrón **mvc** se ve frecuentemente en aplicaciones web, donde la vista es la página **html** y el código que provee de datos dinámicos a la página, el modelo es el **Sistema de Gestión de Base de Datos** y el controlador representa la **Lógica de negocio**.

* *Modelo*: es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos. En **vgc** el modelo se hace en un **Proyecto Java Estándar** que comprende la **lógica de negocio** y los **mapeos de hibernate**.
* *Vista*: este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario. En **vgc** la vista se hace a través de **JSP** dentro de un **Proyecto Web Dinámico**.
* *Controlador*: este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. En vgc el controlador es **Struts**.

#### Struts

Struts es una herramienta de soporte para el desarrollo de aplicaciones web bajo el patrón **mvc** bajo la plataforma **j2ee**. Struts permite reducir el tiempo de desarrollo. Forma parte de la familia **software libre** y su compatibilidad con todas las plataformas en que **java entreprise** esté disponible, lo convierte en una herramienta igualmente disponible.

Cuando se programan aplicaciones web con el patrón mvc, siempre surge la duda de usar un solo controlador o usar varios controladores, pues si consideramos mejor usar un solo controlador para tener toda nuestra lógica en un mismo lugar, nos encontramos con un grave problema, ya que nuestro controlador se convierte en lo que se conoce como **fat controller**. **Struts** surge como la solución a este problema ya que implementa un solo controlador **ActionServlet** que evalúa las peticiones del usuario mediante un archivo configurable “struts-config.xml”.

Los componentes de control son los encargados de coordinar las actividades de la aplicación, que van desde la recepción de datos del usuario, las verificaciones de forma y la selección de un componente del modelo a ser llamado. Por su parte los componentes del modelo envían al control sus eventuales resultados o errores de manera de poder continuar con otros pasos de la aplicación. Entre las características de **struts** se pueden mencionar:

* Configuración del control centralizada.
* Interrelaciones entre acciones y página u otras acciones se especifican por tablas **XML** en lugar de codificarlas en los programas o páginas.
* Componentes de aplicación, que son el mecanismo para compartir información bidireccionalmente entre el usuario de la aplicación y las acciones del modelo.
* Librerías de entidades para facilitar la mayoría de las operaciones que generalmente realizan las páginas **JSP**.
* Contiene herramientas para validación de campos de plantillas bajo varios esquemas que van desde validaciones locales en la página (**javaScript**) hasta las validaciones de fondo hechas a nivel de las acciones.

#### Hibernate

**Hibernate** es una herramienta de mapeo objeto-relacional 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 **xml** que permiten establecer estas relaciones. Es software libre, distribuido bajo los términos de la licencia **GNU LGPL**.

**Hibernate** busca solucionar el problema de la diferencia entre los dos modelos usados hoy en día para organizar y manipular datos: El usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la **OOP**. Luego convertirá los datos entre los tipos utilizados por **java** y los definidos por **SQL** y generará las sentencias **SQL**, liberando al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todas las bases de datos.

**Hibernate** está diseñado para ser flexible en cuanto al esquema de tablas utilizado y para adaptarse a su uso sobre una base de datos existente. También tiene la funcionalidad de crear la **bd** a partir de la información disponible. Ofrece un lenguaje de consulta de datos llamado **HQL**.

### Framework Empleado para Construir el Strategos

Un **framework** es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Este puede incluir soporte de programas, bibliotecas y un lenguaje de **scripting** entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Un **framework** provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. Son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional

Las características más importantes del **framework vgc** son las siguientes:

* Implementa el patrón de desarrollo **MVC**, lo que separa la **vista** el **controlador** y el **modelo** haciendo las aplicaciones más simples de mantener.
* La arquitectura de desarrollo es de 2 capas.
* Lógica de Negocio (**Model**)
* Proyecto Web o Proyecto Cliente/Servidor (**Controller** - **View**)
* Utiliza **Hibernate** como servicio de consultas y persistencias objeto/relacional. Este permite establecer una relación entre una tabla de base de datos con un objeto conceptual. **Hibernate** brinda autonomía con respecto al **SMBDR**.
* Utiliza **Struts** como librería para implementación del **mvc** en proyectos web. Este ofrece control de peticiones y acciones de la aplicación.
* La configuración de las aplicaciones se hace a través de archivos de configuración que tienen extensión “.properties”. Estos archivos de configuración son manejados con una clase estándar de **java**. Por supuesto está particularidad hace las aplicaciones muchos más mantenibles y escalables.
* La implementación de los **servicios de lógica de negocio** se realiza a través de **interfaces**, **clases de implementación** y **factoría**. Gracias a esta filosofía se puede cambiar la implementación sin afectar los módulos que utilicen los **servicios de lógica de negocio**.
* Implementación de clases que representan objetos de sistema con **JavaBeans** simples (con **getters** y **setters**), esta característica permite tener objetos de aplicación más ligeros.
* El manejo de transacción hacia la base de datos se gestiona de manera sencilla.
* Es posible crear varios **servicios de lógica de negocio** para aplicaciones complejas, lo que brinda orden, capacidad de mantenimiento y autonomía de componentes.
* Separación clara de los **servicios de persistencia** dentro de los **servicios lógica de negocio**, lo que facilita el mantenimiento.
* Manejo sencillo de recurso de lenguaje, permite implementación en varios idiomas.
* Implementación de pruebas sencillas dentro del **servicio de lógica de negocio** antes de llevarlo a producción.
* Fácil implementación de reportes gracias a la librería **itext**. Estos reportes son generados en los formatos más ampliamente aceptados **xml**, **pdf**, **html**, etc.
* Incluye un módulo **pre - desarrollado** para gestión de **usuarios**, **grupos** y **permisos**.
* Mecanismos de bloqueo de objetos de sistemas, que permiten concurrencia gestionada y eficiente.
* Registro automático de errores. Estos son cargados en la base de datos de la aplicación que se esté desarrollando.
* Maneja de forma interna todos los parámetros de configuración correspondientes a los usuarios propios de la aplicación.
* La implementación de las vista (**IU**) se realiza con **java server pages**.Dicha tecnología se caracteriza por ser simple y de aprendizaje rápido.

### Cargar la Aplicación en el Ambiente de Desarrollo

**Resumen**

En este capítulo se explicarán los pasos que se debe seguir para sincronizar un aplicación completamente funcional desde **vss**. Se tomará como punto de partida la creación de la base de datos en **postgresql**. Una vez realizada esta actividad, se procederá a crear dos (2) proyectos en eclipse.

1. *Proyecto de lógica de negocio*: en este proyecto java estándar, se realiza el mapeo de objetos en **hibernate** y se crean las **clases** que gestionan la lógica de negocio de la aplicación e independiente del tipo de aplicación que se desee desarrollar (cliente – servidor, web, stand alone, componentes, etc.) Una vez que se han realizado estas clases y todo el mapeo hibernate, se compila la aplicación y se crea un **jar**. Este **jar** se incluirá en el Proyecto web.
2. Proyecto web: contiene todas las páginas **jsp** así como también todos los **action** de **struts** que implementarán los servicios de lógica de negocio creados con anterioridad. Las **jsp** son el mecanismo de interacción con el usuario.

Esta aplicación es un modelo donde el desarrollador puede apreciar en un detalle ligeramente alto, algunas características y el resultado final deseado de forma empírica.

Para este ejemplo se incorporan las prácticas básicas que debe cumplir toda aplicación en **vgc** en cuanto a:

* Estándar en visores, exploradores, buscadores, ficha de datos, etc.
* Se respetan las normas para codificar tales como; comentarios, identación, orden, nomenclatura en el código de cada una de las clases, páginas jsp, archivos xml, etc.
* Desarrollo de aplicaciones en 2 capas.
* Recursos del lenguaje y de configuración de la aplicación, entre otros.

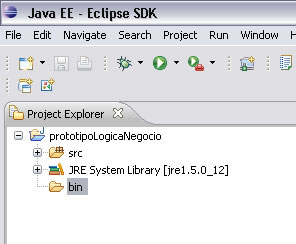
La idea fundamental para este ejemplo práctico es que el lector tenga una visión general del producto final. Una vez que se haya puesto en marcha la aplicación entonces se explicará de forma más concisa **¿cómo se hizo la aplicación?** y cuáles son las consideraciones que se debe respetar al momento de hacer una aplicación en **vgc**.

#### Pasos para la creación de la base de datos

1. Verificar que el servicio de PostgreSQL está iniciado.
2. Abrir el archivo que contiene el script para la creación de la base de datos en la siguiente ruta:
3. Abrir el administrador visual de PostgreSQL **pgadmin**.
4. Cargar el servidor donde se desee crear la base de datos. Puede ser el **Servidor Local**.
5. Crear una base de datos vacía con los siguientes parámetros:
   1. **Nombre**: Strategos
   2. **Owner**: sa
   3. **Encoding**: LATIN1
6. Seleccionar la base de datos que se acaba de crear y abrir el **ejecutor de sentencias sql**.
7. Copiar el script y pegarlo dentro del ejecutor de sentencias. Ejecutar la sentencia.
8. Esta acción crea 17 tablas. 16 tablas “afw” son requeridos por el **framework vgc** y 1 tabla adicional para este ejemplo en particular.
9. Una vez realizados todos estos pasos, se cuenta con una base de datos limpia y con todas las tablas necesarias para la aplicación. De igual manera este script crea la data inicial con la que se levanta una aplicación básica en **vgc**.

#### Pasos para sincronizar el proyecto de lógica de negocio

1. Abrir eclipse.
2. Crear un **Java Project** por el siguiente menú: **File 🡪 New 🡪 Project…** con el **Project name**: StrategosLogicaNegocio y presionar **Finish**.
3. Ya se creó el **Proyecto Java Estándar**. *Ver figura n° 2*.



**Figura n° 2:** proyecto java vacío ya creado en eclipse

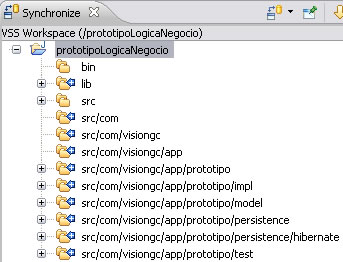
1. Para iniciar el proceso de sincronización desde **vss**, nos paramos sobre el proyecto que se acaba de crear y presionamos clic derecho y seleccionamos el siguiente menú: **Team 🡪 Share Project…**
2. En el primer paso del asistente seleccionamos la opción: **VSS Configuration Wizard** y presionamos **Next >**
3. En el segundo paso del asistente establecemos los siguientes parámetros:
   1. **Username**: [usuario en vss]
   2. **Password**: [password en vss]
   3. **VSSDirectory**: C:\Archivos de programa\Microsoft Visual Studio\COMMON\VSS
   4. **Source dir**:/
   5. **mountpoint**: $/java/aplicaciones/prototipo/version1.0/logicaNegocio

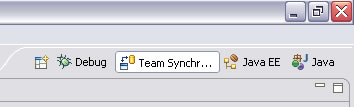
**NOTAS**:

* El usuario de **vss** es el mismo de **Microsoft Windows**.
* Si aún no tiene usuario y password de **vss**, favor contacte con el **administrador** de este servicio.
* Se asume que el archivo “srcsafe.ini” está en la ruta indicada en **VSS Directory** y que este archivo es el correcto.

1. Una vez establecido todos los parámetros, presionamos **Finish**.
2. Nuevamente nos paramos sobre el proyecto y presionamos clic derecho y seleccionamos el siguiente menú: **Team 🡪 Synchronize with the Repository…**
3. Después de realizar esta acción, eclipse cambia de forma automáticamente a la perspectiva de **Team Synchronizing**. *Ver figura n° 3*. Además se muestra el proyecto con todos los directorios que se pueden extraer o sincronizar a nuestro proyecto vacío. *Ver figura n° 4*.

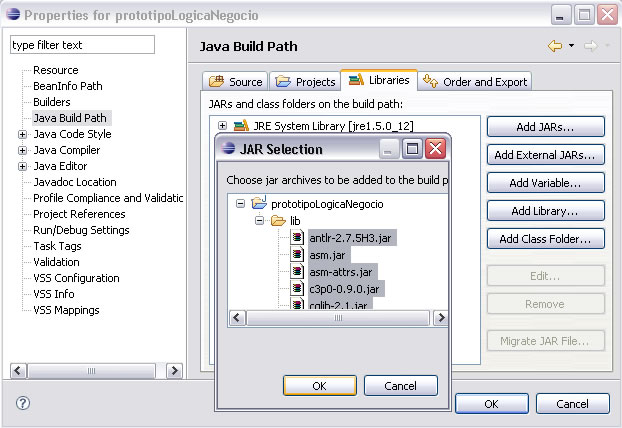
**Figura n° 4:** directorios disponibles





**Figura n° 3:** perspectiva de sincronización

1. Nos paramos sobre el directorio “lib” y hacemos clic derecho y seleccionamos el siguiente menú: 🡪 **Refresh**. Se repite esta acción para el directorio “src”.
2. Después de haber sincronizado los dos directorios anteriores nos cambiamos a la perspectiva de **Java J2EE** nuevamente.
3. Es posible que aparezcan errores en nuestro proyecto. Esto se debe a que tenemos que incluir las librerías de forma explícita a pesar de que fueron sincronizadas con anterioridad.
4. Para incluir las librerías de forma explícita seguimos los siguientes pasos: *Ver figura n° 5*
   1. Hacemos clic derecho sobre el proyecto y seleccionamos el menú: **🡪 Properties**.
   2. En la ventana de propiedad, en el panel izquierdo, seleccionamos la sección de **Java Build Path**.
   3. Cuando hacemos clic sobre dicha sección nos ubicamos en el panel derecho. Seleccionamos la pestaña **Libraries**.
   4. En la parte interna de dicha pestaña presionamos el botón **Add JARs**.
   5. Por defecto aparece nuestro proyecto y dentro de este aparece el directorio “lib”. Dentro de “lib” debemos seleccionar todas las librerías y presionar **Aceptar**.
   6. Finalmente presionamos **OK** de la ventana de **Properties**.



**Figura n° 5:** inclusión de librerías explícitamente

1. Ahora con todos estos pasos ya tenemos el **Proyecto de Lógica de Negocio** completamente listo y sincronizado. Solo resta des-configurar el **vss** para que no ocurran cambios inesperados en el proyecto base de **vss**.

**NOTA**: este paso se hace por seguridad y para evitar cambios en el **Proyecto Base de Lógica de Negocio**. Recuerde que este manual es utilizado por múltiples personas.

1. Nos paramos sobre el Proyecto y hacemos clic derecho y seleccionamos el menú: **Team 🡪 Deconfigure VSS**. Ahora todos los cambios que se realicen en el proyecto son locales y no repercuten en otros proyectos.
2. Ahora creamos una carpeta que nos servirá para albergar el jar de nuestro proyecto. Nos paramos sobre el proyecto, hacemos clic derecho y seleccionamos el menús: **New 🡪 Folder.** A esta carpeta le ponemos el nombre de “jar”.
3. Nos paramos sobre el proyecto, hacemos clic derecho, seleccionamos el menú: **🡪 Refresh**.
4. Nos paramos sobre el proyecto, seleccionamos el menú superior : **Project** **🡪 Clean…**.
5. Ahora debemos crear el “jar” de nuestra aplicación. Para crear el “jar” debemos seguir los siguientes pasos:
   1. Nos paramos sobre el proyecto, seleccionamos el menú: **🡪 Export…**. Dentro de la carpeta “Java” seleccionamos la opción **JAR file** y presionamos **Next >**.
   2. En el siguiente paso, abrimos la raíz del proyecto **prototipoLogicaNegocio**. *Ver figura n° 6*. y configuramos de la siguiente manera:
      1. Nos paramos sobre StrategosLogicaNegocio y desmarcamos “.classpath” y “.project”
      2. Nos paramos en src y marcamos todas las opciones. *Ver figura n° 6*.
      3. El directorio jar lo desactiamos por completo. *Ver figura n° 6*.
      4. El directorio lib lo desactivamos por completo. *Ver figura n° 6*
   3. Seleccionamos el destino de exportación y el nombre:
   4. Presionamos **Finish**.

Ahora tenemos el siguiente jar: “prototipoLogicaNegocio-1.0.0.jar” dentro de la carpeta “jar” del Proyecto de Lógica de Negocio.

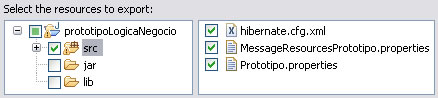
Se desactivan

Sobre el proyecto

Se activan

Sobre el directorio “src”



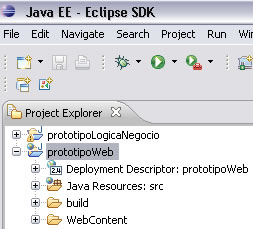


**Figura n° 6:** configuración de archivos de exportación

Estos son todos los pasos que se deben cumpli para tener un **Proyecto Java Estándar de Lógica de Negocio** sincronizado desde **vss** y configurado en el equipo del desarrollador de forma independiente. Además de esto tenemos un archivo **jar**. Este archivo **jar** se incluirá en el **Proyecto Web** y finalmente tendremos una aplicación de 2 capas (**two tiers)** bien definidas.

#### Pasos para sincronizar el Proyecto Web

1. Abrir eclipse.
2. Crear un **Dynamic Web Project** por el siguiente menú: **File 🡪 New 🡪 Project…** con el **Project name**: prototipoWeb y presionar **Finish**.
3. Avanzamos hasta el cuarto paso del asistente, aquí se debe cambiar en parámetro **Java Source Directory**: a JavaSource y luego presionar el botón **Finish**. *Ver figura n° 7*.



**Figura n° 7:** proyecto web vacío ya creado en eclipse

1. Para iniciar el proceso de sincronización desde **vss**, nos paramos sobre el proyecto que se acaba de crear y presionamos clic derecho y seleccionamos el siguiente menú: **Team 🡪 Share Project…**
2. En el primer paso del asistente seleccionamos la opción: **VSS Configuration Wizard** y presionamos **Next >**
3. En el segundo paso del asistente establecemos los siguientes parámetros:
   1. **Username**: [usuario en vss]
   2. **Password**: [password en vss]
   3. **VSSDirectory**: C:\Archivos de programa\Microsoft Visual Studio\COMMON\VSS
   4. **Source dir**: /
4. **Rela. mountpoint**: $/java/aplicaciones/prototipo/version1.0/web
5. Una vez establecido todos los parámetros, presionamos **Finish**.
6. Nuevamente nos paramos sobre el proyecto web, presionamos clic derecho y seleccionamos el siguiente menú: **Team 🡪 Synchronize with the Repository…**
7. Recuerde que después de realizar esta acción, eclipse cambia de forma automáticamente a la perspectiva de **Team Synchronizing**.
8. Nos paramos sobre el directorio “JavaSource” y hacemos clic derecho y seleccionamos el siguiente menú: 🡪 **Refresh**. Se repite esta acción para el directorio “WebContent”.
9. Después de haber sincronizado los dos directorios anteriores nos cambiamos a la perspectiva de **Java J2EE** nuevamente.
10. Ahora con todos estos pasos ya tenemos el **Proyecto Web** completamente listo y sincronizado. Solo resta desconfigurar el **vss** para que no ocurran cambios inesperados en el proyecto base de **vss**.
11. No paramos sobre el Proyecto y hacemos clic derecho y seleccionamos el menú: **Team 🡪 Deconfigure VSS**. Ahora todos los cambios que se realicen en el proyecto son locales y no repercuten en otros proyectos.
12. Debemos incluir el jar que se generó en el Proyecto de Lógica de Negocio, para realizar esto seguimos los siguientes pasos:
    1. Cerrar eclipse.
    2. Ir al directorio ..\..\prototipoLogicaNegocio\jar
    3. Copiar el archivo “prototipoLogicaNegocio-1.0.0.jar”
    4. Ir al directorio ..\..\prototipoWeb\WebContent\WEB-INF\lib\
    5. Pegamos el jar que se copió en el paso anterior en directorio “lib”.
    6. Abrimos eclipse nuevamente.
13. Nos paramos sobre el proyecto, hacemos clic derecho, seleccionamos el menú: **🡪 Refresh**.
14. Nos paramos sobre el proyecto, seleccionamos el menú superior : **Project** **🡪 Clean…**.

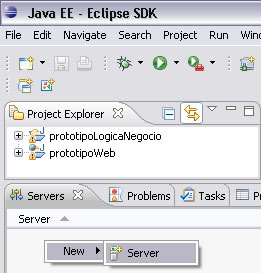
Estos son todos los pasos que se deben cumpli para tener un **Proyecto Web** sincronizado desde **vss** y configurado en el equipo del desarrollador de forma independiente.

#### Pasos para crear una instancia del contenedor de servlet

Se asume que se ha instalado con anterioridad el contenedor de servlet **Apache Tomcat**. Para mayor información, favor consultar *Instalación del ambiente de desarrollo.doc* en el **Kit de Recursos para Desarrolladores**.

En esta sección se va a realizar una instancia del contenedor de servlet **Apache Tomcat**, con el objetivo de que posteriormente podamos ejecutar desde nuestro **ide** la aplicación, para efectos de depuración, corrección, codificación, etc. Los pasos para crea la instancia de Tomcat son los siguientes:

1. Abrir eclipse.
2. Ubicamos la pestaña **Servers** que se encuentra en la parte inferior del ide. *Ver figura n° 8*.
3. En la parte blanca de dicha ficha hacer clic derecho y seleccionar el menú: **New 🡪 Server**.



**Figura n° 8:** creación de instancia de Tomcat

1. En la ventana que se muestra después de ejecutar el paso anterior, se debe configurar los siguientes parámetros:
   1. **Server’s host name**: localhost
   2. **Server type**: Tomcat v5.5. Server
   3. **Server runtime**: Apache Tomcat v5.5.
2. Presionar **Next >**. En el siguiente paso del asistente, se debe seleccionar el proyecto con nombre prototipoWeb y luego presionar el botón **Add**. Esto quiere decir que nuestro **Proyecto Web** lo estamos asociando el Servidor **Tomcat**. Luego presionamos **Finish**.
3. Ahora ya tenemos la instancia de **Tomcat** creada, ahora debemos configurarla para evitar conflictos de puertos a través de los siguientes pasos:
   1. Pinchamos doble clic sobre la instancia que se acaba de crear en la pestaña **Servers**.
   2. Esta acción abre el archivo de configuración de dicha instancia.
   3. Cambiamos los puertos en la tabla de **Ports** de la siguiente manera:
      1. **Tomcat admin port**: 8015
      2. **HTTP**: 8090
      3. **AJP/1.3**: 8019
4. Guardamos los cambios de dicho archivo y lo cerramos.

Estos son todos los pasos que se deben cumpli para crear una instancia de **Tomcat** en nuestro **ide** **eclipse**. Gracias a esta funcionalidad podemos probar más activamente las aplicaciones que estemos desarrollando y así minimizar los tiempos de corrección y depuración.

#### Puesta en marcha de la aplicación

Una vez que se han realizado todos los pasos de las tres secciones anteriores, podemos probar nuestra aplicación desde el **ide**. Para ello se deben seguir los siguientes pasos:

1. Nos paramos sobre el proyecto **prototipoWeb**, hacemos clic derecho sobre este y seleccionamos el menú: **Run As 🡪 Run on Server**.
2. Es posible que muestre una ventana de confirmación para la asociación del proyecto con la instancia de Tomcat. Esta ventana debemos configurar los siguientes parámetros:
   1. **How do you want to select the Server**: Choose an existing server
   2. Activar la casilla Always use this server when running this project.

#### ¿Cómo se hizo el Strategos Advance?

**Conceptos Preliminares**

Antes de explicar ¿cómo se hizo la aplicación prototipo? es conveniente que tengamos claro que toda aplicación debe cumplir una fase de Análisis y Diseño. Para nuestro caso no efectuaremos estas fases, en su defecto listaremos un conjunto de requerimientos exhaustivamente detallados y que delimitan el alcance de la mini aplicación.

1. La aplicación debe implementar los estándares de interfaz de usuarios definidos en el framework de Vision Grupo Consultores.
2. La aplicación debe tener una pantalla donde el usuario se autentique con su respectivo **usuario** y **clave**.
3. La aplicación debe controlar los permisos de los usuarios. No se requiere el desarrollo del módulo de **Gestión de Usuario y Permisos**, pero sí que se validen permisos contra la base de datos.
4. La aplicación debe tener un archivo donde se establezcan los recursos del lenguaje. Esto es para brindar al usuario la posibilidad a futuro de que la aplicación pueda estar en varios idiomas. No es necesario hacerlo en otros idiomas pero sí que todos los textos que se muestran en pantallas sean a través de recursos de lenguajes en español tradicional.
5. La aplicación debe emitir todos los mensajes necesarios en cada pantalla para mantener al usuario informado. Tal es el caso de de mensajes de error, mensajes informativos, etc.
6. Debe tener un menú principal y la **barra de área** debe estar posicionada del lado izquierdo. Esta pantalla es la que se carga una vez se ha autenticado el **usuario**. Debe estar sin ningún elemento en su cuerpo.
7. En la **barra de área** se deben mostrar las siguientes opciones:
   * + Visor tipo lista
     + Visor tipo árbol
8. Debe contener las imágenes en el **encabezado** y en la **barra de aplicación**. Si no conoce alguno de estos elementos, favor consultar *Estándares visuales.doc* en el **Kit de Recursos para Desarrolladores**.
9. En la **barra de aplicación** se debe desplegar las siguientes opciones:
   * + Ir al inicio
     + Manual en línea
     + Acerca de
     + Salir
10. Todas las opciones de la **barra de aplicación** debe estar completamente funcionales, excepto el manual en línea que solo abrirá una ventana vacía.
11. La aplicación debe tener una **barra de navegación** que indique al usuario la posición que se encuentra en cada instante de su interacción con el sistema.
12. Cuando se presione clic sobre la opción del menú principal **Visor tipo lista**, el sistema debe mostrar un **Visor tipo lista**. Este visor debe permitir lo siguiente:
    * + Listar un conjunto limitado de registros.
      + Debe permitir la opción de modificar, eliminar y ver propiedades de cada registro de la lista.
      + Debe permitir la ordenación de registros por cada columna, ascendente y descendentemente y se debe mostrar este criterio en la parte derecha de la **barra inferior** del **contendor forma**.
      + Debe contener un **paginador** que facilite la navegación de la lista y debe estar situado en la parte izquierda de la **barra inferior** del **contendor forma**.
      + En la **barra de menús** del **contenedor forma** se debe colocar la opción de Nuevo que nos facilitar crear un registro.
      + En la **barra de filtros** del **contenedor forma** se debe colocar 2 campos de búsqueda.

**NOTA**: todos estos aspectos deben estar completamente operativos y funcionales y en sincronía con la base de datos.

1. Cuando se presione clic sobre la opción del menú principal **Visor tipo árbol**, el sistema debe mostrar un **Visor tipo árbol** obteniendo sus **nodos** y **raíces** de la base de datos y verificando las dependencias necesarias. Al hacer clic sobre cualquier nodo del **visor tipo árbol**, debe mostrarse en la **barra inferior** del **contenedor forma**.
2. Al momento de presionar clic sobre la opción Nuevo que se encuentra en la **barra de menús** del **contendor forma** se debe mostrar una ficha de datos con el mismo numero de campos que hay en la tabla correspondiente en la base de datos registro, exceptuando el **ID**.
3. De igual manera se debe facilitar modificación y la vista de propiedades en la misma **ficha de datos** y de los mismos campos, claro está, en momentos diferentes y bajo la petición del **usuario** en el **visor tipo lista**.
4. Finalmente se debe permitir la opción de eliminar un registro de la lista.

**NOTA**: CRUD es el acrónimo en inglés de:

**C**: Create

**R**: Read

**U**: Update

**D**: Delete

Es claro que estas son las operaciones fundamentales que se realizan en todo software convencional.

1. Se debe incluir la validación de valores requeridos u opcionales, nulos, numéricos, etc. que sean pertinentes para el **CRUD** de Registros.

Con estos **16 aspectos** se describe de forma detallada y muy plana todos los requerimientos de la **aplicación**. Es claro que los detalles de más bajo nivel deben ser revisados y estudiados por el lector con el fin de comprenderlos y hacer un uso racional del material que se le facilita.

En las siguiente secciones de este capítulo abordaremos los aspectos más resaltante de cómo se hizo la aplicación en ambas capas, **Lógica de Negocio** y **Proyecto Web**. Se debe destacar que por razones de practicidad no será posible escribir un detalle muy elaborado, en tal sentido, es preferible que el desarrollador conozca los pilares de la creación de dichos y que profundice en la práctica.

#### ¿Cómo se hizo el proyecto de lógica de negocio?

Iniciaremos esta sección dando una breve definición de cada uno de las clases e interfaces que componen un paquete básico de **Lógica de Negocio**.

Para iniciar cualquier aplicación de **vgc,** se debe hacer uso expreso y coherente de las **plantillas de proyectos**. Estas **plantillas de proyectos** son aplicaciones que ya están previamente desarrolladas con las características que conforman el esqueleto primordial. Estos recursos permiten que el desarrollador no tenga que iniciar una aplicación desde cero. En un sentido más concreto, **vgc** cuenta con:

* Una plantilla para Proyectos de Lógica de Negocio.
* Una plantilla para Proyectos Web.

Estas plantillas se encuentran en **vss**. Para el caso de un **Proyecto de Lógica de Negocio** como el que se explica en este manual (Lógica de Negocio de la Aplicación Prototipo v.1.0), se tiene que crear un **proyecto java estándar** y posterior a esto se debe sincronizar desde **vss** y desde la siguiente ruta:

$/java/componentes/comunes/vgcLogicaNegocioPlantilla/version1.0

Una vez sincronizado el proyecto de lógica de negocio tenemos los siguientes paquetes y archivos:

**1.- Paquete:** “com.visiongc.plantilla”

* “PlantillaConfiguration.java”: clase que implementa un gestor de parámetros de configuración de aplicación o componente que son almacenadas en un archivo \***.properties** (“Plantilla.porperties”)
* “PlantillaService.java”: interface que define los métodos de negocio que debe suministrar el servicio del framework.

**2.- Paquete:** “com.visiongc.plantilla.impl”

* “PlantillaServiceFactory.java”: clase que implementa una factoría única de servicios de lógica de negocio de framework.
* “PlantillaServiceImpl.java”: clase que implementa la interface de servicios de lógica de negocio del framework.

**3.- Paquete:** “com.visiongc.plantilla.model”

* “ObjetoBD.java”: clase de **java** estilo javaBean OJO que representa un objeto de sistema o del componente.
* “ObjetoBD.hbm.xml”: archivo xml que contiene la configuración de la relación entre el objeto de sistema a nivel de clases de Java y el objeto de sistema a nivel de esquema de datos o base de datos.

**4.- Paquete:** “com.visiongc.plantilla.persistence”

* “PlantillaPersistenceSession.java”: interface que define los métodos de persistencia (consultas, inserciones, modificaciones, eliminaciones) que debe suministrar el servicio de persistencia del framework.
* “PlantillaPersistenceSessionFactory.java”: interface que define los métodos que debe implementar una factoría de servicios de persistencia del framework.

**5.- Paquete:** “com.visiongc.plantilla.persistence.hibernate”

* “PlantillaHibernateSession.java”: clase que implementa la interface de servicio de persistencia del framework utilizando la librería **ORM** Hibernate.
* “PlantillaHibernateSessionFactory.java”: clase que implementa la factoría de servicios de persistencia del framework utilizando la librería **ORM** Hibernate.

Nótese que todos estos paquetes o clases tienen en su estructura verbal la palabra **Plantilla** o **plantilla**, excepto la clase “ObjetoBD.java” y el archivo de configuración de **hibernate** “ObjetoBD.hbm.xml”.

Después de hacer la sincronización del **proyecto base** procedemos a cambiar todos los nombres de los **paquetes** o **clases**, adaptándolos a nuestra aplicación. Cambiamos la palabra **Plantilla** o **plantilla** por **Prototipo** o **prototipo** como se muestra a continuación:

# Paquete 1 #

com.visiongc.app.plantilla # cambiar por 🡪 #

com.visiongc.app.prototipo

# Clase #

PlantillaConfiguration.java # cambiar por 🡪 #

PrototipoConfiguraction.java

# Clase #

PlantillaService.java # cambiar por 🡪 #

PrototipoService.java

# Paquete 2 #

com.visiongc.app.plantilla.imp # cambiar por 🡪 #

com.visiongc.app.prototipo.imp

# Clase #

PlantillaServiceFactory.java # cambiar por 🡪 #

PrototipoServiceFactory.java

# Clase #

PlantillaServiceImpl.java # cambiar por 🡪 #

PrototipoServiceImpl.java

# Paquete 3 #

com.visiongc.app.plantilla.model # cambiar por 🡪 #

com.visiongc.app.prototipo.model

# Clase #

ObjetoBD.java # cambiar por 🡪 #

Registro.java

# Archivo #

ObjetoBD.hbm.xml # cambiar por 🡪 #

Registro.hbm.xml

# Paquete 4 #

com.visiongc.app.plantilla.persistence # cambiar por 🡪 #

com.visiongc.app.prototipo.persistence

# Clase #

PlantillaPersistenceSession.java # cambiar por 🡪 #

PrototipoPersistenceSession.java

# Clase #

PlantillaPersistenceSessionFactory.java # cambiar por 🡪 #

PrototipoPersistenceSessionFactory.java

# Paquete 5 #

com.visiongc.app.plantilla.persistence.hibernate # cambiar por 🡪 #

com.visiongc.app.prototipo.persistence.hibernate

# Clase #

PlantillaHibernateSession.java # cambiar por 🡪 #

PrototipoHibernateSession.java

# Clase #

PlantillaHibernateSessionFactory.java # cambiar por 🡪 #

PrototipoHibernateSessionFactory.java

# Paquete 6 #

com.visiongc.app.plantilla.test # cambiar por 🡪 #

com.visiongc.app.prototipo.test

# Clase #

PlantillaTest.java # cambiar por 🡪 #

PrototipoTest.java

# Archivo #

MessageResourcesPlantilla.properties # cambiar por 🡪 #

MessageResourcesPrototipo.properties

# Archivo #

Plantilla.properties # cambiar por 🡪 #

Prototipo.properties

Estos son todos los pasos previos que se deben realizar para configurar y adaptar nuestra nuevo proyecto de **Lógica de Negocio**. Con respecto al código que se coloca en cada una de estas clases, es importante señalar que este ya viene precargado en la plantilla de **Lógica de Negocio**. Es conveniente que el desarrollador dé un vistazo a cada una de las clases que se encuentran en dicho proyecto, que estudie y analice el código que se encuentra en estas.

Demos entonces, un vistazo general por los pasos que deben cumplir para la codificación de este proyecto:

1. En el archivo “hibernate.cfg.xml” configuramos los parámetros de la base de datos.
2. En el archivo “Prototipo.properties” configuramos los parámetros del proyecto.
3. En el archivo “MessageResourcesPrototipo.properties” configuramos los recursos de lenguaje del proyecto.
4. Crear la clase “Registro.java” donde se escriben todos los **getters** y **setters**. Esta es una clase tipo **javabean** fundamental en este tipo de desarrollo.
5. Crear el archivo “Registro.hbm.xml” donde se hace relación de **hibernate** con la tabla registros de la base de datos en **postgresql**.
6. Implementamos la persistencia con los siguientes pasos:
   1. Creamos la interfaz “PrototipoPersistenceSession.java” y le codificamos los métodos:
      1. getRegistros(); getRegistrosRoot(); getRegistrosHijos();
   2. Crear la clase de implementación de la interfaz anterior con el nombre “PrototipoHibernateSession.java” donde se codifica la implementación del método:
      1. getRegistros(); getRegistrosRoot(); getRegistrosHijos();
7. Hasta este punto tenemos la **persistencia** completa. Ahora implementamos la **lógica de negocio**. Recordemos que solo podemos acceder a la **persistencia** a través de la **lógica de negocio**.

**NOTA**: recordemos que a pesar de poder acceder a los servicios de **persistencia** de forma directa, en **vgc** es una norma que esta práctica no se realice. En este sentido, solo se puede acceder a la **persistencia** a través de la **lógica de negocio**, esto permite unificar la codificación y detectar problemas de manera estructurada.

1. Para implementar el código de la lógica de negocio seguimos los siguientes pasos:
   1. Crear la interfaz “PrototipoService.java” y codificamos tres métodos:
      1. getRegistros( );
      2. deleteRegistro(Registro registro, Usuario usuario);
      3. saveRegistro(Registro registro, Usuario usuario);
      4. getRegistroRoot(boolean recursivo);
      5. refreshArbolRegistros(Registro root, String abiertos, String separador, Long reloadId);
      6. refreshArbolRegistros(Registro root, String abiertos, String separador);
   2. Crear la clase de implementación de la interfaz anterior con el nombre “PrototipoServiceImpl.java” donde se codifica la implementación de los métodos:
      1. getRegistros( );
      2. deleteRegistro(Registro registro, Usuario usuario);
      3. saveRegistro(Registro registro, Usuario usuario);
      4. getRegistroRoot(boolean recursivo);
      5. refreshArbolRegistros(Registro root, String abiertos, String separador, Long reloadId);
      6. refreshArbolRegistros(Registro root, String abiertos, String separador);
2. En este punto ya hemos creado tanto las clases y las interfaces de la **persistencia** como las clases y las interfaces de la **lógica de negocio**. En consecuencia solo resta crear los métodos que nos permiten completar un requerimiento de tipo **CRUD**.

**NOTA**: la aplicación prototipo v.1.0. tiene básicamente las siguientes funcionalidades.

* Mostrar una lista de los registros. Estos se encuentran en una tabla de la base de datos. La tabla se llama **registro** y el componente visual a nivel jsp que los muestra en pantalla es un **visor tipo lista**.
* Insertar un nuevo registro en la tabla **registro** de **postgredql**.
* Modificar un registro ya existente de la tabla **registro** de **postgresql**.
* Eliminar un registro de la tabla **registro** de **postgresql**.

1. Para mayor comprensión de estos pasos es recomendable *Ver el paquete “com.visiongc.app.prototipo” y “com.visiongc.app.prototipo.impl”*
2. Finalmente hacemos **🡪 Refresh** a nuestra aplicación, verificamos que no existan alertas de errores, hacemos clic sobre el menú **🡪 Project 🡪 Clean…** y para culminar esta primera parte creamos el **\*.jar** de la lógica de negocio. La creación del **jar** fue explicada en el capítulo anterior de este manual.
3. El **jar** que se acaba de crear lo debemos incluir en nuestro **Proyecto Web**. El siguiente capítulo hace toda la descripción necesaria sobre la creación del **Proyecto Web**.

#### ¿Cómo se hizo el proyecto web?

Este capítulo hace una explicación sencilla y coherente de los pasos básicos que se deben realizar para crear el **proyecto web**. Recordemos que este proyecto requiere incorporar el **jar** de **lógica de negocio**.

En sentido general el **proyecto web** contiene clases de java así como también todos aquellos archivos que le dan forma y sentido a la aplicación prototipo **v.1.0**, tales como; páginas **jsp**, imágenes en formatos **gif** o **jpg**, archivos de configuración, etc. Al momento de sincronizar el **proyecto web** viene de forma implícita un módulo completamente desarrollado para **Administración de Usuarios y Permisos**. Esta particularidad forma parte del **framework vgc** y al igual que nuestra aplicación contiene páginas jsp, imágenes, etc. pero las clases están en las librerías que se incluyeron desde el **framework** en el proyecto de **lógica de negocio**.

Empecemos con un breve repaso de las cosas que se han realizado hasta este punto durante todo este manual:

1. Se dieron a conocer las definiciones fundamentales del **mvc**, **struts**, **hibernate**, **postgresql** y algunas características del **framework** de **vgc**.
2. Se planteó un diagrama de la arquitectura de la solución **prototipo v.1.0**
3. Se creó una base de datos con las tablas que son requeridas por el **framework** de **vgc** así como una tabla adicional con el nombre de **registro**.
4. Se realizó la sincronización del **proyecto de lógica de negocio** desde **eclipse**.
5. Se creó el **jar** de **lógica de negocio** para incluirlo posteriormente en el **proyecto web**.
6. Se realizó la sincronización del **proyecto web** desde **eclipse**.
7. En ambos proyectos se realizó la configuración de archivos básicos y de la base de datos.
8. En ambos proyectos se explicó la inclusión de librerías de forma explícita.
9. Se creó la instancia de **tomcat** en **eclipse** para ejecutar la aplicación desde el ide.
10. Se puso en marcha la aplicación **prototipo v.1.0**.
11. Se especificaron los requerimientos de la aplicación.
12. Se explicaron cada una de las clases básicas de un **proyecto de lógica de negocio**.
13. Se hizo una explicación de cómo crear el **proyecto de lógica de negocio**.

Partiendo de estos pasos, se asume que el lector ha adquirido madurez en la mayoría de las prácticas y que es capaz de seguir rápidamente los enunciados o pasos que se dictan a continuación.

Para la creación del **proyecto web** se deben seguir los siguientes pasos:

1. Sincronizar el proyecto desde **vss** que se encuentra en la siguiente ruta:

$/java/componentes/comunes/vgcPlantillaWeb/version1.0

1. Eliminar los archivos que nos son necesarios, tal como los de **vss \*.vcc**.
2. Renombrar todos paquetes, pero únicamente se debe reemplazar la palabra **plantilla** por **prototipo**.
3. Renombrar los siguientes archivos, pero únicamente se debe reemplazar la palabra **plantilla** por **prototipo** y respetando las mayúsculas y las minúsculas.
   1. En el directorio **JavaSource**:
      1. “PlantillaMapStrutsActionRight”
      2. “PlantillaWeb.properties”
      3. “plantillaMessageResourcesWeb.properties”
   2. En el directorio **WebContent**:
      1. “struts-plantillaWeb-config.xml”

**NOTA**: en el transcurso del desarrollo de la aplicación se van incorporando archivos, imágenes, páginas, carpetas, etc. El desarrollador debe intuir de forma correcta la ubicación y los nombres de cada uno de estos en función de los archivos similares ya existentes.

1. Configurar los siguientes archivos básicos de la aplicación:
   1. “PrototipoWeb.properties”: en este archivo se debe colocar la ruta completa donde se encuentra la clase de configuración de permisos “PrototipoMapStrutsActionRight”.
   2. “prototipoWebMessageResources.properties”: en este archivo se debe escribir todas las claves y los valores de los elementos de lenguaje.
   3. “web.xml”: en este archivo se debe cambiar el nombre de la aplicación. Colocamos **Prototipo v.1.0**
   4. “struts-prototipoWeb-config.xml”: en este archivo se escribe el mapeo de los **action** que son propios de la aplicación.
   5. “tiles-defs”: en este archivo se configuran las definiciones de layout. En esta aplicación no hace falta cambiarlo.
   6. “validation.xml”: en este archivo se escribe el código de las validaciones que hará struts. En esta aplicación no hace falta cambiarlo.
   7. “validator-rules.xml”: este archivo no se modifica puesto que es distribuido con struts de forma implícita.
2. Crear una página jsp en el siguiente directorio **páginas 🡪 prototipo 🡪 registros** con el siguiente nombre “gestionarRegistrosLista.jsp”.
3. Hacer mapeo **URL** 🡨 🡪 **Action** en el archivo “struts-prototipoWeb-config.xml”. Para mayor información acerca de los pasos antes mencionados vea el *código de la aplicación prototipo v.1.0*
4. Hacer mapeo de permisos en el archivo “PrototipoMapStrutsActionRight”.
5. Crear el **action** (clase java) dentro del paquete “com.visiongc.app.prototipo.web.struts.actions” con el nombre “GestionarRegistrosListaAction.java”.
6. Configuración del menú de la aplicación.

Una vez que se han completado todos estos pasos, el desarrollador debe hacer **🡪 Refresh**  y luego hacer las validaciones pertinentes para que la aplicación no genere errores. Posterior a esto es recomendable hacer **🡪 Project 🡪 Clean…** del proyecto. Con todos estos pasos ya es posible iniciar la aplicación.

#### Glosario de Términos Empleados

**Framework**: traducción al inglés de la palabra marco de trabajo. Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto

**Front End**: es la parte del software que interactúa con el usuario. Hace referencia a la visualización del navegador.

**IU**: acrónimo de interfaz de usuario. Es la forma en que los usuarios pueden comunicarse con una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo

**JSP**: acrónimo de Java Server Pages. Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

**IDE**: un entorno de desarrollo integrado o en inglés **I**ntegrated **D**evelopment **E**nvironment es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

**JAVA**: una tecnología desarrollada por Sun Microsystems para aplicaciones software independiente de la plataforma, que engloba:

* Lenguaje de programación Java
* API Java
* Máquina Virtual de Java (JVM)
* Applet Java
* Java Runtime Environment

**JDK**: el **J**ava **D**evelopment **K**it, por sus siglas en inglés, es un compilador y conjunto de herramientas de desarrollo para la creación de programas independientes y applets java.

**VSS**: **M**icrosoft **V**isual **S**ourceSafe es una herramienta de Control de versiones que forma parte de Microsoft Visual Studio.

**MVC**: acrónimo del Model View Controller. Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

**PHP**: acrónimo de Preprocessor Hyper Text. Es un lenguaje de programación usado normalmente para la creación de contenido para sitios web con los cuales se puede programar las páginas **html** y los códigos de fuente

**Tag**: traducción al inglés de la palabra marca. Las marcas son utilizadas en lenguajes como **html** para definir la estructura de un determinado documento.

**Wizard**: en desarrollo de aplicaciones, es un elemento de interfaz de usuario que permite segmentar un proceso en pasos más simples, con el objeto de dar claridad y sencillez en la interacción con el usuario.

# Capítulo V Estándares Visuales

Este capítulo muestra la descripción de los estándares de los **visores**, **buscadores, exploradores, fichas de datos, y contenedores,** empleados en la construcción del Strategos Advance. Los aspectos contemplados para explicar el estándar son los siguientes:

* + - Nombre, tipo y descripción de los elementos.
    - Estilo a utilizar para cada elemento en cuestión.
    - Uso de los atributos en los **tags** **html**.
    - Ejemplos gráficos de la apariencia final de los elementos a estudiar.

Los estándares visuales que se explican en la siguiente documentación son y deben ser independientes del **framework** de desarrollo adoptado por **Visión Grupo Consultores C.A.** Esta característica permite tener aplicaciones altamente uniformes donde el control de calidad es la base para producir software robusto, configurable, mantenible, funcional y con simplificación en la capa visual.

Los estándares del manual se muestran en el lenguaje universal para páginas y desarrollo web, **html** con uso de hojas de estilos (**css).**

## Convenciones Utilizadas

Con el objeto de mejorar el rendimiento en la compresión de ideas en esta documentación, se crearon algunas convenciones que pondrán alerta al lector con respeto a lo que ocurre en el transcurso de la implementación. Tales convenios son los siguientes:

**NOTA**: Los recuadros sombreados con la indicación de **NOTA** muestran información de importancia y que el lector deberá recordar y que es sumamente relevante dentro del contexto que se introduce.

*El estilo cursiva se utiliza para comentarios al margen del tema que está siendo tratado.*

Los estilo del texto:

* + - Al introducirlas, **destacamos** en negrita las palabras importantes.
    - Mostramos las combinaciones de teclado así: *Control+A* o *Shift+Supr*
    - Mostramos los nombres de los archivos o carpetas así: “nombreArchivo” o “nombreCarpeta”
    - Mostramos los códigos y líneas de comando en el texto así: código() o línea de comando
    - Se debe respetar el uso de altas y bajas al momento de leer la información.

En este recuadro se escribirán los comandos y códigos que deben ser tipeados de forma idéntica bien sea en bloc de notas o cualquier editor de páginas web.

Los texto rojos que se encuentran entre numerales no deben ser escrito en el código, estos solo son explicativos

# Comentario de ejemplo #

<!—Comentario de ejemplo -->

.estilo {

color:black;

font-family:Verdana;

font-size:12px;

otro:12xx

}

## Componentes

Cuando se desarrollan aplicaciones de tipo empresarial, se tiene un conjunto de elementos que, desde una perspectiva visual, permiten llevar a cabo las tareas requeridas para el cual fueron diseñadas e incluidas en el software.

Unificaremos algunos conceptos antes de entrar en materia de estándar, y que nos permiten contextualizar los controles (componentes) respecto a su funcionalidad.

Cuando nos referimos a **componentes** hablamos de un conjunto de controles, objetos o elementos visuales agrupados que aparecen en las distintas páginas de la aplicación que se desarrolla, con el propósito de realizar la funcionalidad propia del software y sus requerimientos. En este ámbito, el concepto de **componente** no es igual que en la programación orientada a objetos u orientada a componentes. Un **componente** pueden ser dos cajas de textos, un contenedor, una barra de navegación, etc. Para efectos de este manual se han categorizado los componentes de forma tal que, el lector los pueda identificar a lo largo de la lectura. Algunos ejemplos son:

* **Visores**: contiene imágenes, encabezado, cuerpo, filas, columnas, etc.
* **Exploradores**: contiene imágenes, nodos, filas, jerarquías, etc.
* **Buscadores**: contiene cajas de textos, imágenes, etiquetas, etc.
* **Fichas de datos**: contiene cajas de textos, etiquetas, listas desplegables, cuadros de opción, etc.

### Contenedores

Se les llama **contenedor** a los componentes que están diseñados con el fin único de tener dentro de su cuerpo a otro componente, objeto o control. Los **contenedores** son la base del estándar, puesto que, en ellos se diseñan o colocan todos aquellos botones, vínculos, imágenes, controles, etc. que son iguales en todas las pantallas del sistema. Los **contenedores** facilitan la ubicación de los componentes albergados en su cuerpo, y simplifica en gran medida el trabajo del desarrollador al momento de codificar e implementar estos estilos. Los **contenedores** se diseñan de forma tal que sean reutilizables y configurables, esta característica dependerá del **líder de desarrollo** y sus habilidades para aislar la escritura excesiva de código a través de artificios o frameworks que brinden esta peculiaridad.

A pesar de que hay una diversidad de contenedores desde al punto de vista aplicativo, básicamente se tienen (2) dos tipos de **contenedores** enlas aplicaciones que se desarrollan en **Visión Grupo Consultores C.A.** estos son los siguientes:

* Contenedor Principal
* Contenedor Forma

El **contenedor principal** es la base de toda la aplicación, mientras que, el **contenedor forma** está contenido dentro del **contenedor principal**. Por lo general y por convención del equipo de desarrollo, dentro del **contenedor forma**, se deben mostrar todos los componentes, objetos y controles de la aplicación. Esto quiere decir, que; siempre es necesario programar el **contenedor principal** y en el cuerpo de este se debe programar el **contenedor forma**. El cuerpo del **contenedor forma** va a depender del requerimiento que se esté modelando.

**NOTA**: los únicos elementos que no se rigen por esta norma son los **reportes**, pero no se contemplan en esta documentación.

### Visores

Es un elemento de la interfaz de usuario (**IU**) que permite visualizar un conjunto de datos de forma cómoda y sencilla. La presentación de los datos depende del requerimiento y el diseño realizado por el **analista de sistemas**. En **Visión Grupo Consultores** **C.A.** tenemos visores de 2 tipos:

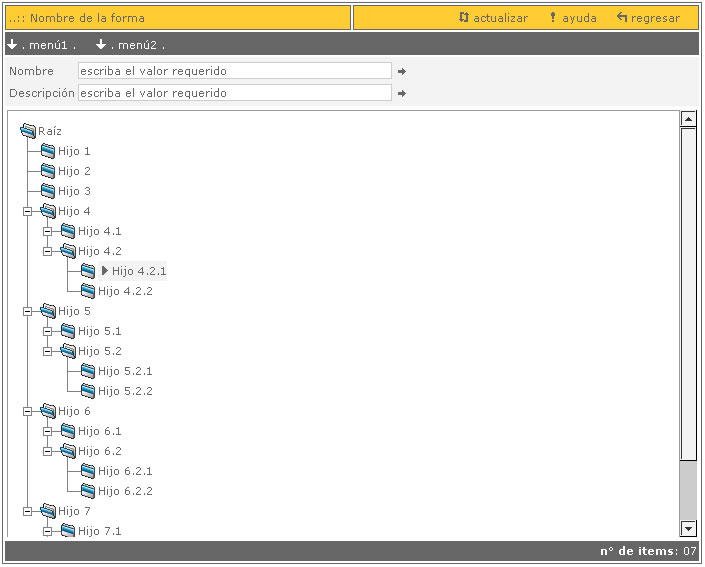
* **Árbol**: para este caso se utiliza una graficación tipo **treeview** con **n** niveles de ramificaciones
* **Lista plana**: para este caso se utiliza una graficación tipo **listview** con un solo nivel de orden.

En cualquiera de los casos, la función primordial de estos es brindar una vista consolidada de registros y que están asociadas a operaciones diversas como, agregar, modificar, asociar, eliminar, etc.

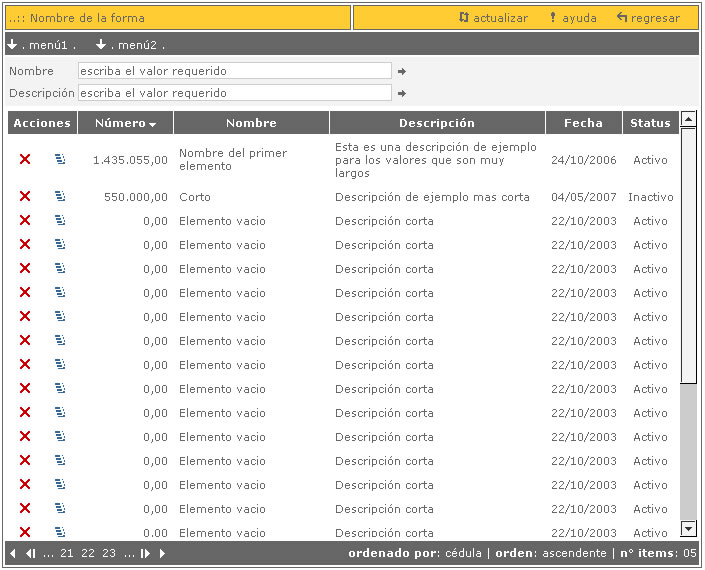
En los **visores tipo árbol** tenemos registros con un solo campo visible. Este campo visible puede ser la concatenación de dos o más atributos de un objeto o de simples strings. *Ver figura n° 1*

En los **visores tipo lista** tenemos un conjunto de registros que se muestran en una tabla de al menos una columna. *Ver figura n° 2*

**NOTA**: estos son los casos más simples de **visores tipo árbol** y **visores tipo lista**. Más adelante se explicará las variaciones de estos, dependiendo de su funcionalidad y contexto.



**Figura n° 1:** visor tipo árbol



**Figura n° 2:** visor tipo lista

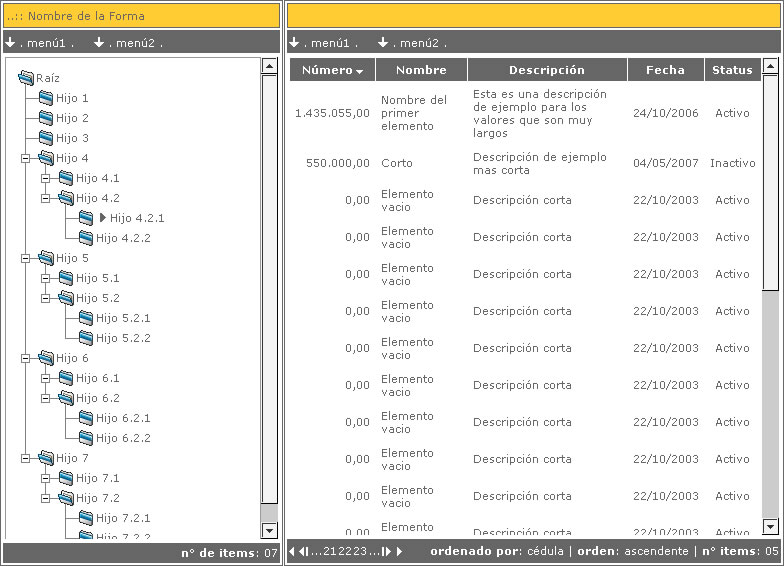
### Exploradores

Se puede definir como un conjunto de visores agrupados convenientemente, que cumplen funciones más especializadas que los visores señalados anteriormente. Es importante hacer una correcta definición de estos elementos con el propósito de unificar el lenguaje y aumentar la comunicación entre **analistas**, **programadores** y **consultores**. Los **exploradores** pueden estar formados por **visores tipo árbol** o por **visores tipo lista**, esto va a depender de la funcionalidad del requerimiento.

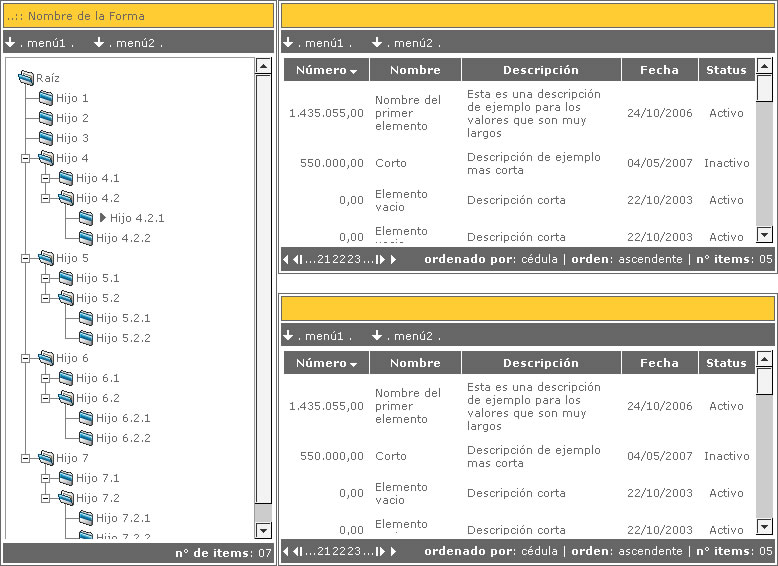
Como su propio nombre lo indica su función primordial es explorar, para tal fin es necesario tener elementos de apoyo que faciliten la exploración. Los **exploradores** al igual que los **visores** están asociados a las operaciones de agregar, modificar, asociar, eliminar, imprimir, etc.

Básicamente podemos tener (2) dos tipos de **exploradores**:

* Explorador treeview + listview *Ver figura n°3*
* Explorador treeview + doble listview *Ver figura n° 4*



**Figura n° 3:** explorador treeview + listview



**Figura n° 4:** explorador treeview + doble listview

### Buscadores

Son elementos auxiliares en todo el software. Su función básica es la de seleccionar registros en una vista consolidada como puede ser una lista plana o de árbol. Los **buscadores** también son conocidos dentro del lenguaje de **Visión Grupo Consultores C.A.** como **selectores** o **browsers**.

Si estudiamos estos conceptos en profundidad podemos apreciar que puede existir confusión e inclusive ambigüedad con respecto a los **visores** o más aún con los **exploradores**. Para aclarar esta situación *Ver tabla n° 1* donde se aprecia la visión completa entre componentes, nombre / sinónimos y tipo de elemento.

**NOTA**: los buscadores, tal y como describe la definición anterior, son elementos complementarios y de apoyo, que son invocados a partir de una pantalla principal y a su vez por un objeto que indique de manera intuitiva que; a partir de ese botón, se hará la llamada al **buscador** propiamente dicho.

Ejemplo: Sin título-2 esta imagen, generalmente aparece en las **fichas de datos**. Se puede invocar un buscador haciendo clic sobre botón y debe aparecer de forma modal el elementosolicitado.

**Componente Nombre/Sinónimo Tipo IU**

Visor lista Visor / listview Primario

Visor árbol Visor / treeview Primario

Explorador árbol + lista Explorador / simple Primario

Explorador árbol + doble lista Explorador / doble Primario

Buscador lista Buscador / selector / browser / listview Secundario/Auxiliar

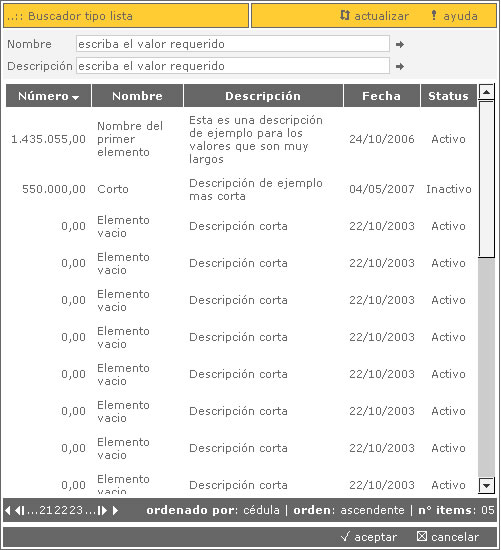
Buscador árbol Buscador / selector / browser / treeview Secundario/Auxiliar

**Tabla n° 1**: relación entre componente, nombre / sinónimo y tipo de interfaz de usuario

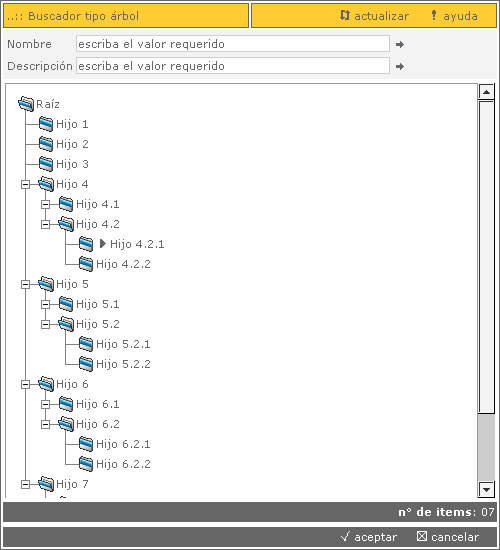
Básicamente podemos tener (2) dos tipos de **buscadores**:

* Buscador tipo lista *Ver figura n°5*
* Buscador tipo árbol *Ver figura n° 6*

**Figura n° 5:** buscador tipo lista



En la figura se puede apreciar claramente que son similares a los visores y a los exploradores. Fundamentalmente los buscadores se basan en visores. Solo algunos pequeños objetos o controles son añadidos o eliminados para dar la funcionalidad final y deseada al componente buscador.



**Figura n° 6:** buscador tipo árbol

En esta primera parte de la documentación se definieron algunos conceptos fundamentales de desambiguación, los cuales debe tener en cuenta el desarrollador al momento de implementar los capítulos siguientes. La idea es conciliar conceptos y aclarar el panorama general para desarrollo de aplicaciones en **Visión Grupo Consultores C.A.** Se puede apreciar la similitud entre los **visores**, **exploradores** y los **buscadores**. En términos generales el componente realmente base es el **visor**, de este, se hereda el mismo estilo visual y hacemos juego de varios componentes, teniendo de esta forma como resultado los a los **exploradores**. El **buscador** también hereda el estilo visual del **visor**, con la particularidad de que los **buscadores** son ventanas de apoyo y deben ser mostrados en forma tal que se pueda captar este tipo de funcionalidad. El buscador finalmente debe devolver un ítem o varios de la lista que se mostró en forma modal.

Es evidente el uso de lineamientos para la programación de todos los componentes propios de cualquier aplicación. Estos aplican a todos los objetos, controles que se deseen implementar en las aplicaciones comunes.

### Detalle de los Componentes Visuales del Strategos Advance

En la mayoría de las aplicaciones que se desarrollan, se utilizan componentes similares y hasta es posible que sean iguales. Estos componentes son los pilares fundamentales de las aplicaciones para **vgc**.

Es conveniente mencionar, que estos componentes comunes no son el universo de elementos que se utilizan en las aplicaciones de **vgc**. Esto quiere decir que; en la medida que se requiera implementar un nuevo componente, entonces el equipo de desarrollo lo crea, luego lo estandariza y finalmente se pone en práctica.

A continuación se muestra una lista de estos componentes comunes sin orden de valor o importancia.

* **Bienvenida**: es la primera página de la aplicación. Esta página se utiliza como un artificio para poder cargar de forma automática a la página de autenticación. La página de **bienvenida** hace una llamada a través de una función **javascript** llamada window.open( ). Gracias a esta instrucción se puede controlar el tamaño y la ubicación de la aplicación
* **Autenticación**: permite al usuario autenticarse en la aplicación. Se solicita usuario y clave de acceso. Esta página es similar para casi todas las aplicaciones.
* **Menú principal**: es el componente raíz de toda la funcionalidad fundamental de la aplicación. Este muestra las distintas opciones en la barra de navegación. Dichas opciones se categorizan dependiendo de los requerimientos del software y del modelamiento del analista. La barra de área puede estar situada tanto en la izquierda como en la parte superior del contenedor principal.
* **Mensajes**: son todos aquellos textos que mantienen comunicación constante y prudente con el usuario final. Esta comunicación se realiza de forma informativa, preventiva, a modo de excepción o en caso de errores, facilitando la toma de decisiones al usuario y protegiendo los datos contra operaciones indebidas. Toda aplicación debe tener mensajes.
* **Lanzadores**: permiten servir de filtro o armadores previos para poder ejecutar alguna acción deseada. En este tipo de elementos lo importante es la estandarización de los diversos objetos o controles que se muestran dentro de ellos. Así como también deben ser alineados los artificios de programación que implemente el desarrollador.
* **Visores**: componente que permite visualizar datos de forma ordenada. *Ver capítulo V*
* **Exploradores**: componente que permite visualizar datos de forma ordenada en varios visores agrupados convenientemente .*Ver capítulo VI*
* **Buscadores**: componente auxiliar que permite visualizar datos de forma ordenada y que devuelve uno o varios registros de la lista seleccionados y que se pueden utilizar desde el componente primario que lo invocó. *Ver capítulo VII*
* **Fichas de datos**: componente que permite efectuar las operaciones fundamentales de agregación, edición de datos de la base datos. *Ver capítulo VIII*
* **Barra de progreso**: son aquellos componentes que permiten al usuario saber cuándo debe esperar a que se ejecute una acción determinada en el software. De igual manera, le informan qué porcentaje de avance aproximado lleva un proceso o actividad del sistema. Es muy útil puesto que sin este tipo de componentes el usuario no daría un uso correcto al sistema, recargando y forzando la ejecución de muchos procesos que se van cargando lentamente en la memoria del computador.

Iniciaremos nuestra definición de estándares desde los componentes más genéricos hasta llegar a los componentes que son más pequeños. Es necesario abordar el tema de acuerdo a la siguiente estructura, puesto que este es el orden en que se desarrollan cada uno de los componentes y es el mismo orden de encapsulamiento que debe considerar el **líder de desarrollo**:

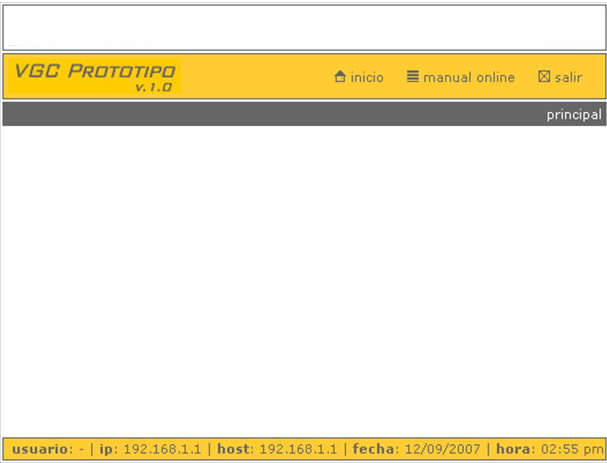
* + - Contenedor principal
* Contenedor forma
* Visores
* Exploradores
* Fichas de datos
* Buscadores

Como se mencionó con anterioridad, este manual solo hace referencia precisa y explicita sobre los componentes listados anteriormente. Estos son la base para cualquier aplicación empresarial en **Visión Grupo Consultores C.A.** Es posible que existan otro componentes nuevos y por hacer, pero probablemente estos estén basados en su mayoría en los que se definen en esta documentación.

Es importante recalcar que los estilos y diseño visual de estos componentes es perfectible y pueden variar dependiendo de la decisión del Gerente de Proyecto. El uso de colores es referencial y pretende ser una guía base para poder abordar cualquier tipo de aplicación. En cualquiera de los casos, el Gerente de Proyecto debe conciliar con el equipo de desarrollo, todos los estilos y consideraciones necesarias al momento de iniciar cualquier tipo de proyecto.

**NOTA**: la comunicación del equipo en cada proyecto debe ser eficaz y correcta. Cualquier situación de cambio o mejora a las aplicaciones en el aspecto visual es muy bien aceptada, solo se debe informar y conciliar con los involucrados para evitar retrabajo y pérdida de esfuerzo.

**a) Partes del contenedor principal**: *Ver figura n° 7*



Barra superior (encabezado)

Barra de aplicación

Barra de navegación

Cuerpo

**Figura n° 7:** partes del contenedor principal

Barra de estado

* **Barra superior (encabezado)**: en esta barra por lo general se colocan imágenes, textos o animaciones que identifiquen al cliente. Para el caso de los entes gubernamentales ya está diseñado el cintillo obligatorio con el que deben cumplir. Esto no restringe la posibilidad de editar y adecuar esta primera barra superior a conveniencia del cliente.
* **Barra de aplicación**: esta barra muestra los botones y vínculos propios de la aplicación. Esta puede adecuarse previa conciliación entre **Visión Grupo Consultores C.A.** y el **Cliente**. Como se puede apreciará la parte izquierda de la **barra de aplicación** puede contener textos, imágenes o animaciones que identifican al **Producto**. En la parte derecha de la barra están los botones que el **Cliente** no puede cambiar o adaptar. En cualquiera de los casos la adaptación de la **barra de aplicación** está sujeta a decisión exclusiva de **Visión Grupo Consultores C.A.**
* **Barra de navegación**: esta barra es de uso estándar para todas las aplicaciones en **Visión Grupo Consultores C.A.** no puede ser cambiada o eliminada. En esta se muestran los puntos válidos de navegación. El usuario final puede ir a cualquier punto de navegación que aparezca en esta barra haciendo clic sobre los textos que allí se muestran. La aplicación internamente realiza las validaciones necesarias sobre qué puntos deben aparecer y cuáles no.
* **Cuerpo**: en esta sección debe aparecer el **contenedor forma** tal y como se mencionó en el capítulo anterior. El cuerpo es la parte donde se incrustan otros componentes, objetos o controles dependiendo del requerimiento y la funcionalidad de la aplicación.
* **Barra de estado**: esta barra es de uso estándar para todas las aplicaciones en **Visión Grupo Consultores C.A.** no puede ser cambiada o eliminada. En esta se muestran valores de información general a la aplicación, tales como; usuario, dirección ip, host, fecha, hora. Es posible incluir otro tipo de información en esta barra pero está sujeta a decisión exclusiva de **Visión Grupo Consultores C.A.**

**b.) Márgenes del contenedor principal**: *Ver figura n° 8*

Navegador / IE / Mozilla

0 px

0 px, border:1px

Barra superior

Contenedor Principal

0 px, border:1px

0 px

30 px

Barra de aplicación

30 px

0 px, border:1px

0 px, border:1px

0 px, border:1px

Barra de navegación

Cuerpo

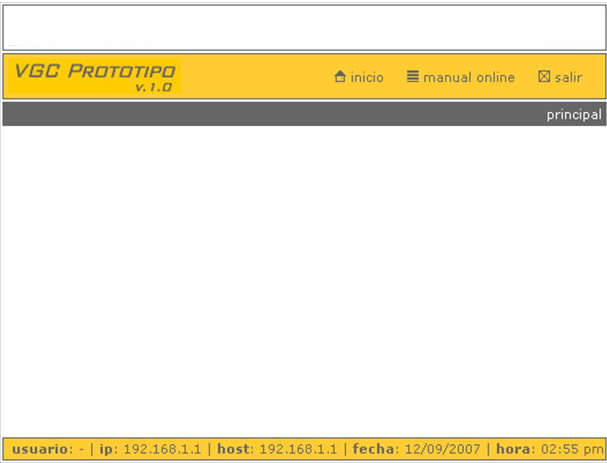
0 px, border:1px

Barra de estado

20 px

**Figura n° 9:** márgenes del contenedor principal

**c.) Acotación del contenedor principal**: *Ver figura n° 10*



30 px.

30 px.

Definido por el texto

Es relativa

20 px.

100 % (Es relativa y dependiente del tamaño del browser)

**Figura n° 10:** acotación del contenedor principal

* El cuerpo es la diferencia que hay entre la sumatoria de todas las barras y el tamaño del browser.

Cuerpo = (30px + 30px + tamaño barra de navegación + 20px) - tamaño browser

* El tamaño de la barra de navegación depende del tipo de fuente utilizada para el estilo. Por lo general la fuente utilizada es de 11px. (font size:11px.) equivalente a 20px.

**d.) Código de implementación del contenedor principal**: a continuación se muestra el código **HTML** que sebe ser programado para crear el contenedor principal en su forma más simple. Para probar este código realice los siguientes pasos:

1.- Crear una página web vacía con cualquier herramienta web.

2.- Enlazar la página con la hoja de estilo.

3.- Copiar todo el código y pegarlo entre las marcas <body></body> y guardar los cambios.

4.- Visualizar la página o refresca el navegador y listo.

<!-- Esta es la "Barra de Navegación del Contenedor Principal" -->

<tr>

<td colspan="2" class="barraNavegacionPrincipal">&nbsp;</td>

</tr>

<!-- Este es el "Cuerpo del Contenedor Principal" -->

<tr>

<td colspan="2">

&nbsp;

</td>

</tr>

<!-- Esta es la "Barra de Estado del Contenedor Principal" -->

<tr>

<td class="barraInferiorPrincipal" colspan="2">

&nbsp;

</td>

</tr>

</table>

<!-- Este es el "Contenedor Principal" -->

<table class="contenedorPrincipal">

<!-- Esta es la "Barra Gubernamental del Contenedor Principal" -->

<tr>

<td align="center" class="barraGubernamentalPrincipal" colspan="2">

<table align="left" >

<tr>

<td>&nbsp;</td>

</tr>

</table>

<table border="0" align="right" height="100%">

<tr>

<td>&nbsp;</td>

</tr>

</table>

</td>

</tr>

<!-- Esta es la "Barra Superior del Contenedor Principal" -->

<tr>

<td align="center" class="barraSuperiorPrincipal" colspan="2">

<table align="left">

<tr>

<td>&nbsp;</td>

</tr>

</table>

<table border="0" align="right" height="100%">

<tr>

<td>

&nbsp;

</td>

</tr>

</table>

</td>

</tr>

**d.) Estilos que implementan el contenedor principal**: a continuación se muestran los estilos que implementa cada una de las partes del contenedor principal. En caso de que se desee adecuar algún elemento visual se debe hacer a través de la hoja de estilos.

* **Barra superior**: a continuación de muestra el estilo de esta sección.

/\* Barra superior del contenedor principal \*/

.barraGubernamentalPrincipal {

height:30px;

border-style:solid;

border-width:1px;

border-color:#666666;

background-color:#FFFFFF;

}

* **Barra de aplicación**: a continuación de muestra el estilo de esta sección.

/\* Barra de aplicación del contenedor principal \*/

.barraSuperiorPrincipal {

height:30px;

border-style:solid;

border-width:1px;

border-color:#666666;

background-color:#FFCC33;

}

* **Barra de navegación**: a continuación de muestra el estilo de esta sección.

/\* Barra de navegación del contenedor principal \*/

.barraNavegacionPrincipal {

height:0px;

border-style:solid;

border-width:1px;

border-color:#666666;

padding:3px;

background-color:#666666;

font-size:11px;

color:#FFFFFF;

font-family:Verdana;

text-decoration:none;

text-align:right;

}

* **Barra de estado**: a continuación de muestra el estilo de esta sección.

/\* Barra inferior del contenedor principal\*/

.barraInferiorPrincipal {

height:20px;

border-style:solid;

border-width:1px;

border-color:#666666;

background-color:#FFCC33;

font-family:Verdana;

font-size:11px;

color:#666666;

text-align:right;

}

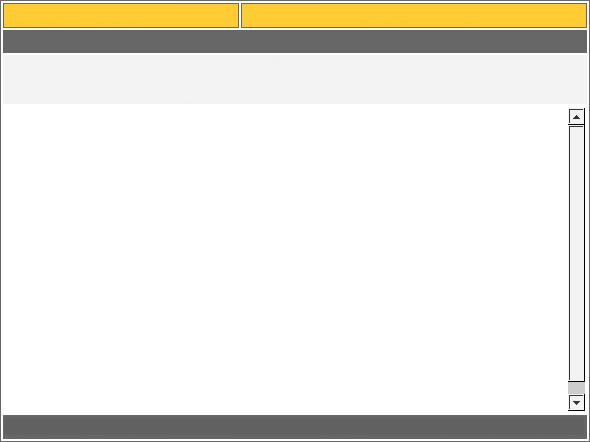
#### Contenedor forma

**a.) Vista general del contenedor forma**: *Ver figura n° 11*



**Figura n° 11:** vista general del contenedor forma

**b.) Partes del contenedor forma**: *Ver figura n° 12*



Barra superior (encabezado)

Barra de menús

Barra de filtros

Cuerpo

Barra inferior (de estado)

**Figura n° 12:** partes del contenedor forma

* **Barra superior (encabezado)**: esta barra está subdividida en dos columnas iguales. La parte izquierda debe llevar el nombre de la forma, mientras que en la parte derecha se debe colocar los botones convenientes dependiendo del tipo de componente que se muestra en el cuerpo del contenedor forma. En los siguientes capítulos se plantean casos reales con componentes específicos.
* **Barra de menú**: esta sección es de uso exclusivo para la colocación de menús. Los menús son desplegables y en ellos aparecen las diversas acciones que se pueden ejecutar con el componente contenido en el cuerpo del contenedor forma. Esta barra es opcional, dependiendo del componente contenido. En los siguientes capítulos se plantean casos reales con componentes específicos.
* **Barra de filtros**: en esta sección de colocan todos aquellos controles u objetos que de alguna manera u otra indiquen pre-filtrado de los elementos que se muestran en el cuerpo del contenedor forma. Por lo general se colocan cuadros de texto para indicar que hay algún campo precargado o para hacer búsqueda por el criterio deseado. Esta barra es de uso opcional, dependiendo del componente contenido. En los siguientes capítulos se plantean casos reales con componentes específicos.
* **Cuerpo**: en esta sección se incrustan otros componentes, objetos o controles dependiendo del requerimiento y la funcionalidad de la aplicación. Es importante recalcar que, la barra de menús, la barra de filtros y la barra inferior están estrechamente ligada al componente que se muestra en el cuerpo, por lo tanto hay que estar atento a su implementación.
* **Barra inferior (de estado)**: en esta barra permite colocar los botones pertinentes de acuerdo al tipo de objeto o control contenido. El uso de esta barra es obligatorio, y se debe estar atento al tipo de componente contenido para así incrustar los botones, paginadores y elementos en general que sean necesarios. En los siguientes capítulos se plantean casos reales con componentes específicos.

**c.) Márgenes del contenedor forma**: *Ver figura n° 13*

Navegador / IE / Mozilla

0 px

0 px, border:1px

Barra superior

Contenedor Principal

0 px, border:1px

0 px

30 px

Barra de aplicación

30 px

0 px, border:1px

0 px, border:1px

0 px, border:1px

Barra de navegación

Cuerpo

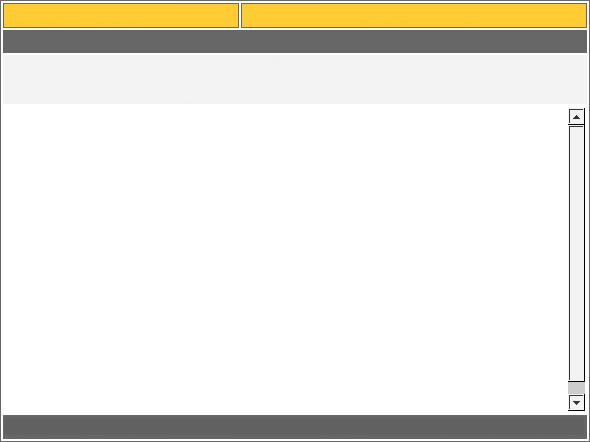
0 px, border:1px

Barra de estado

20 px

**Figura n° 13:** márgenes del contenedor forma

**d.) Acotación del contenedor forma**: *Ver figura n° 14*



25 px.

20 px.

Definido por el texto

Es variable

20 px.

100 % (Es relativa y dependiente del componente contenido)

**Figura n° 14:** acotación del contenedor forma

El alto del cuerpo es establecido por el programador y depende en su totalidad del tipo de componente que sea incrustado.

* El tamaño de la barra de navegación depende del tipo de fuente utilizada para el estilo. Por lo general la fuente utilizada es de 11px. (font size:11px.) equivalente a 20px.

**e.) Código de implementación del contenedor forma**: a continuación se muestra el código **HTML** que sebe ser programado para crear el contenedor principal en su forma más simple. Para probar este código realice los siguientes pasos:

1.- Realizar los pasos del **1** al **3** del **capítulo III** del contenedor principal.

2.- Copiar todo el código y pegarlo dentro del cuerpo del contenedor principal.

3.- Guardar los cambios.

4.- Visualizar la página o refresca el navegador y listo.

<!-- Este es el "Contenedor Secundario o Forma" -->

<table class="contenedorForma" cellspacing="2" cellpadding="2" width="700px">

<!-- Barra Superior del "Contenedor Secundario o Forma" -->

<tr>

<td class="barraSuperiorForma">..:: Nombre de la forma</td>

<td class="barraSuperiorForma" align="right" width="50%">&nbsp;</td>

</tr>

<!-- Barra de Menús del "Contenedor Secundario o Forma" -->

<tr class="barraMenusForma">

<td colspan="2">

&nbsp;

</td>

</tr>

<!-- Barra de Filtros del "Contenedor Secundario o Forma" -->

<tr class="barraFiltrosForma">

<td colspan="2">

&nbsp;

</td>

</tr>

<!-- Cuerpo del "Contenedor Secundario o Forma" -->

<tr>

<td colspan="2">

&nbsp;

</td>

</tr>

<!-- Barra Inferior del "Contenedor Secundario o Forma" -->

<tr class="barraInferiorForma">

<td colspan="2">

&nbsp;

</td>

</tr>

</table>

**f.) Estilos que implementan el contenedor forma**: a continuación se muestran los estilos que implementa cada una de las partes del contenedor forma. En caso de que se desee adecuar algún elemento visual se debe hacer a través de la hoja de estilos.

* **Barra superior (encabezado)**: a continuación de muestra el estilo de esta sección.

/\* Barra superior de la forma \*/

.barraSuperiorForma {

height:25px;

border-style:solid;

border-width:1px;

border-color:#666666;

background-color:#FFCC33;

font-size:11px;

color:#666666;

font-family:Verdana;

}

* **Barra de menús**: a continuación de muestra el estilo de esta sección.

/\* Barra de menús de la forma \*/

.barraMenusForma {

height:20px;

background-color:#666666;

}

* **Barra de filtros**: a continuación de muestra el estilo de esta sección.

/\* Barra de filtros de la forma \*/

.barraFiltrosForma {

background-color:#F3F3F3;

font-family:Verdana;

font-size:11px;

color:#666666;

}

* **Barra de estado**: a continuación de muestra el estilo de esta sección.

/\* Barra inferior de la forma \*/

.barraInferiorForma {

height: 20px;

background-color: #666666;

text-align: right;

font-size: 11px;

color: #FFFFFF;

font-family: Verdana;

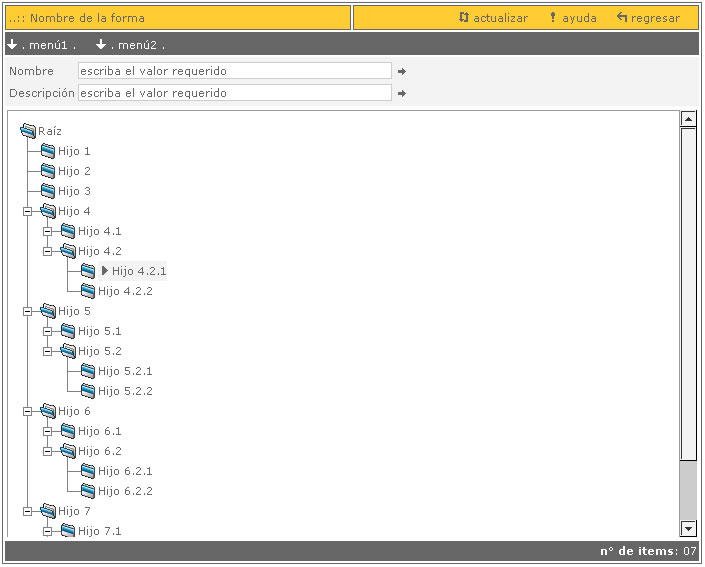
text-decoration: none;

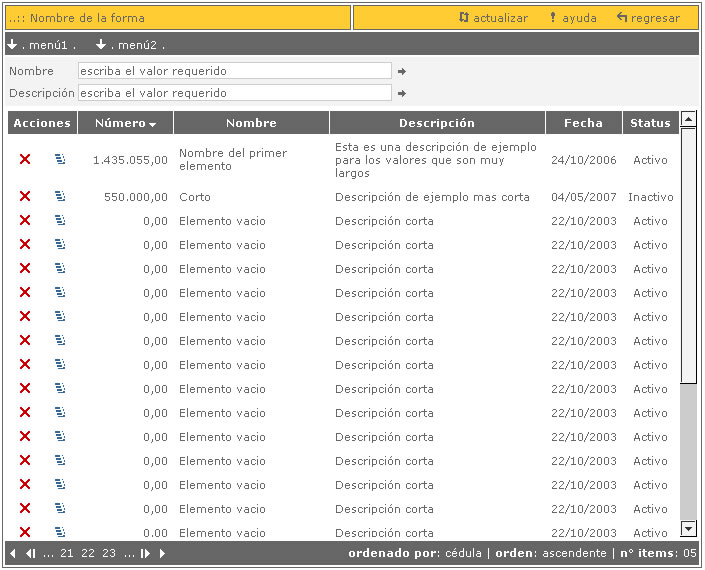
}

#### Visores

**a.) Vista general de visores tipo árbol y tipo lista**: *Ver figura n° 15 y 16*

Las figuras de visores que se muestran a continuación están referenciadas dentro del contenedor forma y del contenedor principal respectivamente. Ambos contenedores fueros explicados en los capítulos anteriores.





**Figura n° 16:** visor tipo lista

**Figura n° 15:** visor tipo árbol

**b.) Partes del visor**: las partes del visor dependen del tipo que sea implementado.

Partes del Visor tipo lista: *Ver figura nº 17*

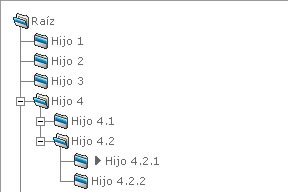


Encabezado

Registros

**Figura n° 17:** partes del visor tipo lista

Partes del visor tipo árbol: *Ver figura nº 18*



Nodo

Cuerpo

**Figura n° 18:** partes del visor tipo árbol

* **Encabezado**: esta sección aplica solo para el visor tipo lista. El encabezado indica el nombre de las columnas, estas a su vez vínculos, permiten establecer la forma de ordenación de los registros. Solo se puede ordenar por una columna a la vez de forma ascendente o descendente. Al lado de la columna de ordenación hay una imagen que indica que dicha columna es el criterio de ordenación e igualmente indica si está ascendente o descendentemente.
* **Registros**: esta sección aplica solo para el visor tipo lista. En esta parte aparecen todos los registros con correspondencia coherente al encabezado. Al momento de implementar la rutina que lista los registros se debe considerar el efecto de animación que se le dan a las filas.
* **Nodos**: esta sección aplica solo para el visor tipo árbol. Esta parte es simple, puesto que muestra un conjunto de elementos de forma jerárquica. Esta jerarquía tiene solo una raíz. Las imágenes utilizadas permiten visualizar las dependencias entre registros.

**NOTA**: al momento de implementar los visores se debe tener en cuenta que este componente estrictamente debe adecuar tres 3 secciones del contenedor forma. Estas secciones son:

* Barra de menú.
* Barra de filtros
* Barra de estado o inferior.

Para los visores que no requiera alguna de estas secciones, simplemente se deberá omitir el código del contenedor forma.

**c.) Código de implementación del visor**:

A continuación se muestra el código **HTML** que sebe ser programado para crear un **visor tipo lista** en su forma más simple. Copie y pegue este código en el cuerpo del contenedor forma.

<!-- Este es el "Visor Tipo Lista" -->

<table class="listView" cellpadding="5" cellspacing="1">

<!-- Este es el Encabezado del "Visor Tipo Lista" -->

<tr class="encabezadoListView">

<td align="center" colspan="2"><b>Acciones</b></td>

<td align="center" width="20%"><b>Número</b><img onClick="javascript:ordenarTabla(this);" style="cursor:pointer" name="imagenOrdenar" src="../imagenes/listView/abajo.gif" border="0" width="10" height="10" alt="Ascendente"></td>

<td align="center" width="30%"><b><a href="#" onMouseOver="this.className='mouseEncimaEncabezadoListView'" onMouseOut="this.className='mouseFueraEncabezadoListView'" class="mouseFueraEncabezadoListView">Nombre</a></b></td>

<td align="center" width="40%"><b><a href="#" onMouseOver="this.className='mouseEncimaEncabezadoListView'" onMouseOut="this.className='mouseFueraEncabezadoListView'" class="mouseFueraEncabezadoListView">Descripción</a></b></td>

<td align="center" width="10%"><b><a href="#" onMouseOver="this.className='mouseEncimaEncabezadoListView'" onMouseOut="this.className='mouseFueraEncabezadoListView'" class="mouseFueraEncabezadoListView">Fecha</a></b></td>

<td align="center" width="10%"><b><a href="#" onMouseOver="this.className='mouseEncimaEncabezadoListView'" onMouseOut="this.className='mouseFueraEncabezadoListView'" class="mouseFueraEncabezadoListView">Status</a></b></td>

</tr>

<!-- Este es el Cuerpo del "Visor Tipo Lista" -->

<tr class="mouseFueraCuerpoListView" onMouseOver="this.className='mouseEncimaCuerpoListView'" onMouseOut="this.className='mouseFueraCuerpoListView'">

<td align="center"><img onClick="javascript:ordenarTabla(this);" style="cursor:pointer" src="../imagenes/listView/eliminar.gif" border="0" width="10" height="10" alt="Eliminar"></td>

<td align="center"><img onClick="javascript:ordenarTabla(this);" style="cursor:pointer" src="../imagenes/listView/propiedades.gif" border="0" width="10" height="10" alt="Propiedades"></td>

<td align="right">1.435.055,00</td>

<td>Nombre del primer elemento</td>

<td>Esta es una descripción de ejemplo</td>

<td align="center">24/10/2006</td>

<td align="center">Activo</td>

</tr>

</table>

A continuación se muestra el código **HTML** que sebe ser programado para crear un **visor tipo árbol** en su forma más simple. Copie y pegue este código en el cuerpo del contenedor forma.

<!-- Este es el "Visor Tipo Árbol" -->

<table class="bordeTreeView" cellpadding="0" cellspacing="10">

<tr><td>

<table class="treeView" border="0" cellpadding="0" cellspacing="0">

<tr>

<td><img src="../imagenes/treeView/topeexp.gif" border="0" width="20" height="20"></td>

<td onMouseOver="this.className='mouseEncimaCuerpoTreeView'" onMouseOut="this.className='mouseFueraCuerpoTreeView'" onDblClick="javascript:alert('Aquí se ejecuta la acción corresponsiente')" class="textoCuerpoFuera" colspan="9">Raíz</td>

</tr> <tr>

<td><img src="../imagenes/treeView/empalme.gif" border="0" width="20" height="20"></td>

<td><img src="../imagenes/treeView/nodo.gif" border="0" width="20" height="20"></td>

<td onMouseOver="this.className='mouseEncimaCuerpoTreeView'" onMouseOut="this.className='mouseFueraCuerpoTreeView'" onDblClick="javascript:alert('Aquí se ejecuta la acción corresponsiente')" class="textoCuerpoFuera" colspan="8">Hijo 1</td>

</tr>

<tr>

<td style="cursor:pointer"><img src="../imagenes/treeView/menosempal.gif" border="0" width="20" height="20"></td>

<td><img src="../imagenes/treeView/nodoexp.gif" border="0" width="20" height="20"></td>

<td onMouseOver="this.className='mouseEncimaCuerpoTreeView'" onMouseOut="this.className='mouseFueraCuerpoTreeView'" onDblClick="javascript:alert('Aquí se ejecuta la acción corresponsiente')" class="textoCuerpoFuera" colspan="8">Hijo 2</td>

</tr>

</table>

</td></tr>

</table>

**f.) Estilos que implementa los visores**: a continuación se muestran los estilos que implementa cada una de las partes de los visores. En caso de que se desee adecuar algún elemento visual se debe hacer a través de la hoja de estilos.

* **Tabla del visor tipo lista**: a continuación de muestra el estilo de esta sección.

/\* List view \*/

.listView {

width:100%;

font-size:11px;

color:#FFFFFF;

font-family:Verdana;

text-decoration:none;

}

* **Encabezado del visor tipo lista**: a continuación de muestra el estilo de esta sección.

/\* Encabezado del list view \*/

.encabezadoListView {

font-family:Verdana;

font-size:11;

color:#FFFFFF;

text-decoration:none;

background-color:#666666;

}

* **Efectos sobre el cuerpo del visor tipo lista**: a continuación de muestra el estilo de esta sección.

/\* Evento over del cuerpo del list view \*/

.mouseEncimaCuerpoListView {

background-color:#F3F3F3;

cursor:pointer;

color:#666666;

}

/\* Evento out del cuerpo del list view \*/

.mouseFueraCuerpoListView {

background-color:#FFFFFF;

color:#666666;

}

* **Tabla del visor tipo árbol**: a continuación de muestra el estilo de esta sección.

/\* Tabla principal del visor tipo árbol \*/

.treeView {

background-color:#FFFFFF;

font-family:Verdana;

font-size:11px;

color:#666666;

}

* **Efectos en los nodos del visor tipo árbol**: a continuación de muestra el estilo de esta sección.

/\* Evento over del cuerpo del tree view \*/

.mouseEncimaCuerpoTreeView {

background-color:#F3F3F3;

cursor:pointer;

color:#666666;

}

/\* Evento out del cuerpo del tree view \*/

.mouseFueraCuerpoTreeView {

background-color:#FFFFFF;

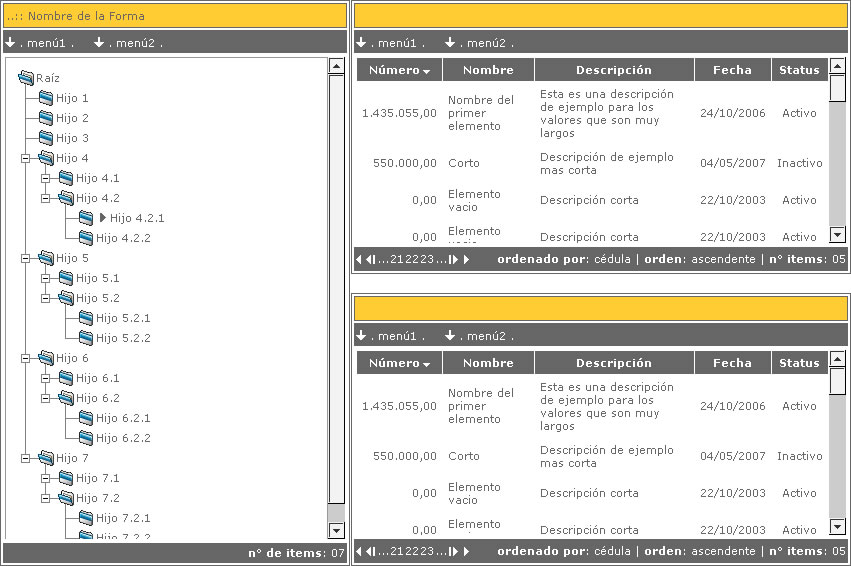
color:#666666;

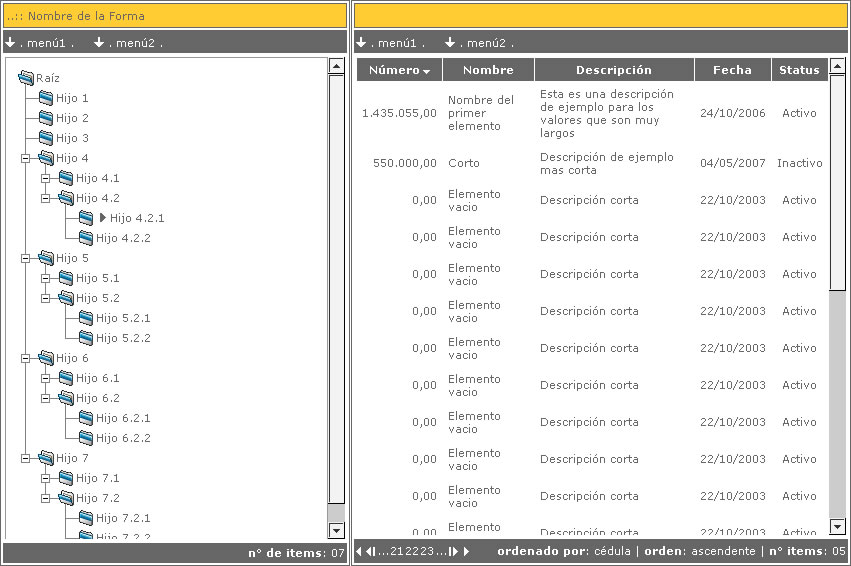
}

#### Exploradores

**a.) Vista general de exploradores**: *Ver figura n° 19 y 20*

Las figuras de exploradores que se muestran a continuación están referenciadas dentro del contenedor forma y del contenedor principal respectivamente. Ambos contenedores fueros explicados en los capítulos anteriores.





**Figura n° 20:** explorador árbol + doble lista

**Figura n° 19:** explorador árbol + lista

**b.) Partes de un explorador**: estos componentes están formados por dos partes. *Ver figura nº 20*



Parte Izquierda

Parte Derecha

**Figura n° 21:** partes de un explorador

* **Parte izquierda**: esta sección tiene incrustada un visor tipo árbol con todas las características y partes propias de este componente. En otras palabras, la parte izquierda del explorador no es algo nuevo, más bien es la reutilización del componente visor tipo árbol. Las dimensiones y características de esta parte izquierda dependerán del requerimiento que se está modelando.
* **Parte derecha**: en esta sección se incrusta uno o dos visores tipo lista. Al igual que en la parte izquierda, el visor tipo lista que se introduce en la parte derecha del explorador debe respetar y considerar todas las características propias de este componente. Para el caso de que se tenga un solo visor tipo lista incrustado, este debe tener el tamaño adecuado y dependiente del requerimiento. Para el caso de que se incrusten dos visores tipo lista, se deben colocar uno debajo del otro y con un ancho igual, respetando simetría entre ambos, mas sin embargo, el alto de cada visor tipo lista dependerá del juicio del analista que modele el requerimiento.

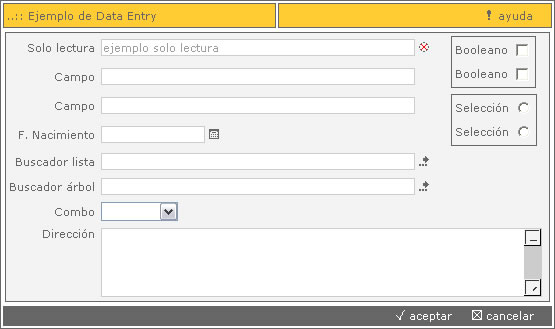
**NOTA**: para este componente no se hace la descripción del código y estilos que se implementan, debido a que estos son heredados de los visores. Los visores se explicaron en el capítulo anterior.

En sentido general podemos afirmar que los exploradores no tienen estilos propios. Estos son la agrupación de varios visores tipo árbol o lista, por supuesto con su respectivo **contenedor forma**.

#### Fichas de Datos

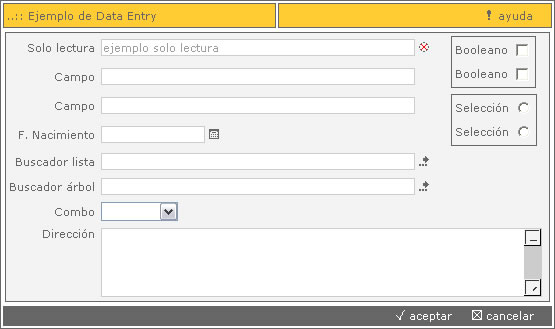
**a.) Vista general de la ficha de datos**: *Ver figura n° 22*

Las figuras de exploradores que se muestran a continuación están referenciadas dentro del contenedor forma y del contenedor principal respectivamente. Ambos contenedores fueros explicados en los capítulos anteriores.



**Figura n° 22:** ficha de datos

**b.) Partes de una ficha de datos**: la ficha de datos es un conjunto de controles agrupados dentro del cuerpo de un contenedor forma. Esa ficha de datos cuenta con una sección que es propiamente el cuerpo del contenedor forma. Es importante señalar que la ficha de datos debe adaptar la barra inferior del contenedor forma, de forma tal que, se incrusten los botones de aceptar y cancelar. *Ver figura nº 23*



Cuerpo

Controles contenidos

**Figura n° 23:** partes de una ficha de datos

* **Cuerpo**: esta sección tiene incrustada todos aquellos controles que permitan cubrir la funcionalidad propia de los requerimientos. En el cuerpo se deben colocar de forma ordenada todos estos controles, respetando márgenes tanto con los bordes del cuerpo del contenedor forma como los márgenes entre cada tipo de control. Es conveniente hacer agrupaciones de controles del mismo para dar orden a la interfaz.

**c.) Controles contenidos en una ficha de datos**:

* **Cuadro de texto**: es un control que forma parte de la ficha de datos. Su función es facilitar al usuario la escritura de textos, números, caracteres, etc., para así interactuar en sentido coherente con el software.
  + Vista previa del cuadro de texto:

Sin título-1

**Figura n° 24:** vista previa del cuadro de texto

* + Código de implementación del cuadro de texto:

/\* Cuadro de texto \*/

<input type="text" class="cuadroTexto" size="50">

* + Estilos que implementa el cuadro de texto:

/\* Cuadro de texto \*/

.cuadroTexto {

font-weight:normal;

font-size:11px;

color:#666666;

font-family:Verdana;

text-decoration:none;

border-style:solid;

border-width:1px;

border-color:#CCCCCC;

}

* **Cuadro de texto solo lectura**: este control es idéntico al cuadro de texto visto anteriormente, con la particularidad que no se puede modificar. Su función es mostrar datos bien sea para dar referencia al usuario o para restringir la edición o manipulación de algunos campo bajo condiciones controladas, etc. Este control tiene en la parte derecha, una imagen que indica que no es editable; esto con el propósito de alertar al usuario y hacer la interfaz más intuitiva.
  + Vista previa del cuadro de texto solo lectura:

Sin título-3

**Figura n° 25:** vista previa del cuadro de texto solo lectura

* + Código de implementación del cuadro de texto solo lectura:

/\* Cuadro de texto solo lectura \*/

<input type="text" disabled="disabled" class="cuadroTexto" size="50" value="ejemplo solo lectura">

<img src="../imagenes/bloqueado.gif" border="0" width="10" height="10" alt="elemento bloqueado">

* + Estilos que implementa el cuadro de texto solo lectura:

/\* Cuadro de texto \*/

.cuadroTexto {

font-weight:normal;

font-size:11px;

color:#666666;

font-family:Verdana;

text-decoration:none;

border-style:solid;

border-width:1px;

border-color:#CCCCCC;

}

* **Etiqueta**: este control forma parte de la ficha de datos. Su función es mostrar describir de forma precisa y corta a los distintos controles que se encuentran en las fichas de datos, tal es el caso de los cuadros de textos, cuadros combinados, botones de selección simple, botones de selección múltiple. Este control debe estar alineado a la izquierda de la ficha de datos y en armonía con los demás controles de la ficha de datos.
  + Vista previa de la etiqueta con cuadro de texto solo lectura:

Sin título-4

**Figura n° 26:** vista previa de la etiqueta con cuadro de texto

* + Código de implementación de la etiqueta:

/\* Etiqueta \*/

<td align="right">Solo lectura</td>

**NOTA**: la etiqueta no implementa estilo alguno. Esta hereda el estilo que tiene el cuerpo del contenedor forma.

Es importante la correcta implementación de todos los controles y componentes a los largo del desarrollo del software, puesto que el mal uso puede causar inconsistencias no deseadas en el producto final.

* **Cuadro combinado**: este control tiene como función principal, permitir al usuario seleccionar un elemento de una lista finita y predefinida. Este comportamiento es útil para dar flexibilidad y control del lado del usuario. Los elementos que aparecen en la lista son controlados internamente por el software. Es posible configurar este tipo de controles para que se puedan seleccionar varios elementos de la lista, pero esa explicación escapa del alcance de este manual.
  + Vista previa del cuadro combinado:



**Figura n° 27:** vista previa del cuadro combinado

* + Código de implementación del cuadro combinado:

/\* Cuadro combinado \*/

<select class="cuadroCombinado">

<option> </option> <option>Opción 1</option> <option>Opción 2</option> <option>opción 3</option> <option>etc.</option>

</select>

* + Estilos que implementa el cuadro combinado:

/\* Cuadro combinado \*/

.cuadroCombinado {

font-weight:normal;

font-size:11px;

color:#666666;

font-family:Verdana;

text-decoration:none;

border-style:solid;

border-width:1px;

border-color:#CCCCCC;

}

* **Cuadro de texto memo**: este control tiene las mismas características que el cuadro de texto, con la particularidad que se utiliza para escribir textos largos, comúnmente conocido como memo. Estos controles deben tener barras de desplazamiento internas y deben respetar el alto propio del control.
  + Vista previa del cuadro de texto memo:



**Figura n° 28:** vista previa del cuadro de texto memo

* + Código de implementación del cuadro de texto memo:

/\* Cuadro de texto memo \*/

<textarea cols="70" rows="5" class="cuadroTexto"></textarea>

* + Estilos que implementa el cuadro de texto memo:

/\* Cuadro de texto \*/

.cuadroTexto {

font-weight:normal;

font-size:11px;

color:#666666;

font-family:Verdana;

text-decoration:none;

border-style:solid;

border-width:1px;

border-color:#CCCCCC;

}

* **Botón calendario**: este control es una imagen que se utiliza para invocar al componente calendario. Este control siempre debe acompañar a un cuadro de texto solo lectura en la parte derecha. Este botón se usa para colocar fechas de forma sencilla en el cuadro de texto solo lectura asociado.
  + Vista previa botón calendario con cuadro de texto:

Sin título-7

**Figura n° 29:** vista previa del botón calendario

* + Código de implementación del botón calendario con etiqueta y cuadro de texto solo lectura:

/\* Botón calendario con etiqueta y cuadro de texto solo lectura \*/

<td align="right">F. Nacimiento</td>

<td colspan="2"><input type="text" class="cuadroTexto" disabled="disabled" size="15"> <img style="cursor:pointer" src="../imagenes/calendario.gif" border="0" width="10" height="10" alt="seleccionar registro"></td>

* + Estilos que implementa el botón calendario:

/\* Botón calendario \*/

style="cursor:pointer"

width="10"

height="10"

/\* El botón es una imagen y sus estilo se colocan en los atributos del Tag html \*/

* **Botón buscador**: este control es una imagen que se utiliza para invocar al componente buscador. Este control siempre debe acompañar a un cuadro de texto solo lectura en la parte derecha.
  + Vista previa botón buscador etiqueta y cuadro de texto solo lectura:

Sin título-8

**Figura n° 30:** vista previa del botón buscador

* + Código de implementación del botón buscador con etiqueta y cuadro de texto solo lectura:

/\* Botón buscador con etiqueta y cuadro de texto solo lectura \*/

<td align="right">Buscador lista</td>

<td colspan="2"><input type="text" class="cuadroTexto" disabled="disabled" size="50"> <img style="cursor:pointer" src="../imagenes/selector.gif" border="0" width="10" height="10" alt="invocar buscador tipo lista" onClick="abrirBuscadorA()"></td>

* + Estilos que implementa el botón buscador:

/\* Botón calendario \*/

style="cursor:pointer"

width="10"

height="10"

/\* El botón es una imagen y sus estilo se colocan en los atributos del Tag html \*/

* **Selección simple**: son botones de radio que permiten selecciona un elemento de un conjunto de estos agrupados. Estos botones deben estar contenidos en una tabla con borde simple
  + Vista previa de selección simple con botones de radio:

Sin título-9

**Figura n° 31:** vista previa selección simple

* + Código de implementación selección simple con botones de radio:

/\* Selección simple \*/

<table class="contenedorBotonesSeleccion">

<tr>

<td>Selección</td> <td><input type="radio" name="1"

class="botonSeleccionSimple"></td>

</tr>

<tr> <td>Selección</td>

<td><input type="radio" name="1"

class="botonSeleccionSimple"></td>

</tr>

</table>

* + Estilos que implementa selección simple con botones de radio:

/\* Selección simple \*/

style="cursor:pointer" width="10" height="10"

/\* El botón es una imagen y sus estilo se colocan en los atributos del Tag html \*/

/\* Botones de radio o selección simple \*/

.botonSeleccionSimple {

border-color:#CCCCCC;

}

* **Selección múltiple**: son botones de chequeo que permiten seleccionar varios elementos de un conjunto de estos agrupados. Estos botones deben estar contenidos en una tabla con borde simple
  + Vista previa de selección simple con botones de chequeo:

Sin título-10

**Figura n° 31:** vista previa selección múltiple

* + Código de implementación selección múltiple con botones de chequeo:

/\* Selección múltiple \*/

<table class="contenedorBotonesSeleccion">

<tr>

<td>Booleano</td>

<td><input type="checkbox"

class="botonSeleccionMultiple"></td>

</tr>

<tr>

<td>Booleano</td>

<td><input type="checkbox"

class="botonSeleccionMultiple"></td>

</tr>

</table>

* + Estilos que implementa selección múltiple con botones de chequeo:

/\* Selección múltiple \*/

style="cursor:pointer" width="10" height="10"

/\* El botón es una imagen y sus estilo se colocan en los atributos del Tag html \*/

/\* Botones de selección múltiple \*/

.botonSeleccionMultiple {

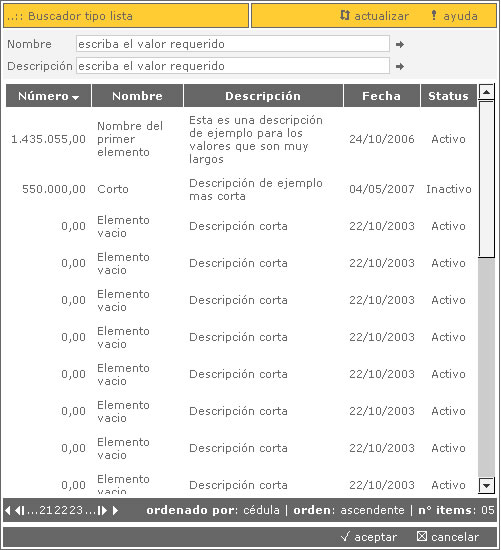
border-color:#CCCCCC;

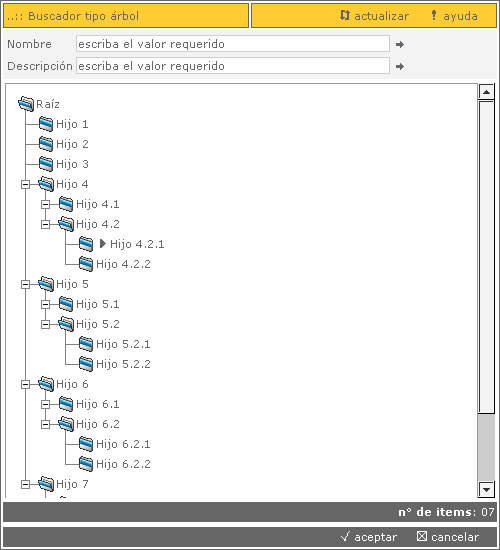
}

#### Buscadores

**a.) Vista general de buscadores**: *Ver figura n° 32 y 33*

Las figuras de buscadores que se muestran a continuación están referenciadas dentro del contenedor forma y del contenedor principal respectivamente. Ambos contenedores fueros explicados en los capítulos anteriores.





**Figura n° 33:** buscador tipo árbol

**Figura n° 32:** buscador tipo lista

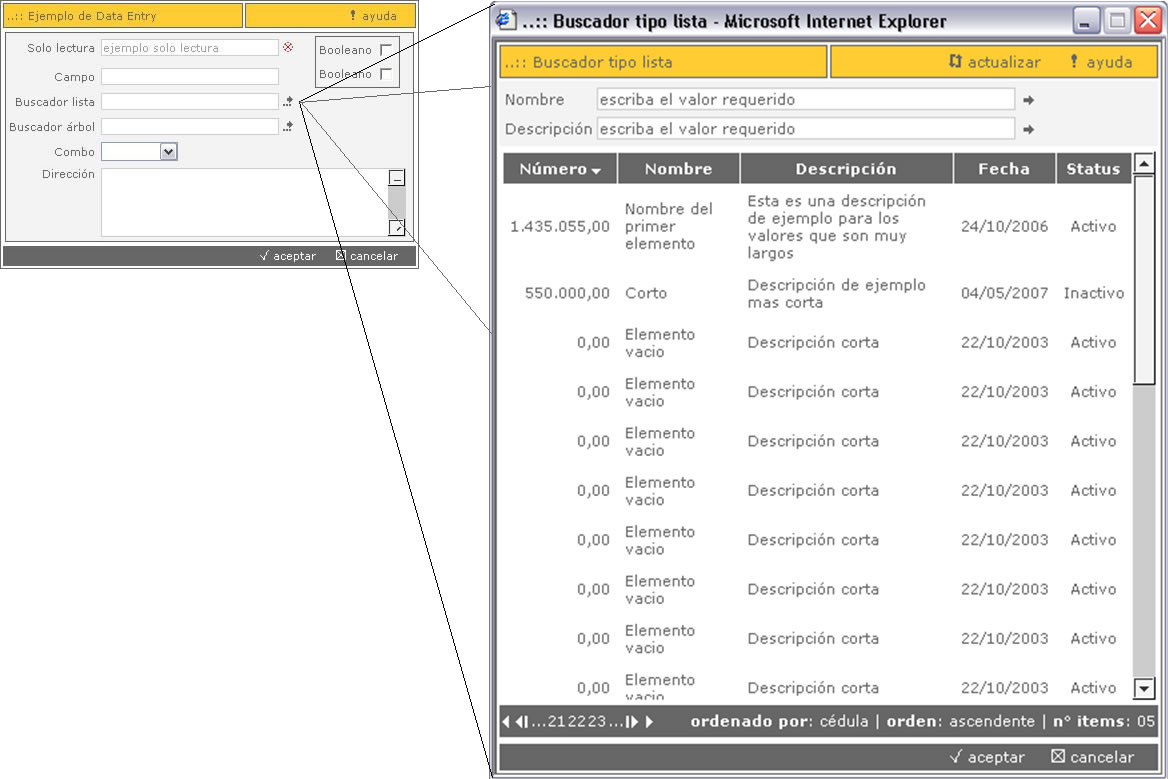
**b.) Partes de un buscador**: las partes del buscador son exactamente iguales a las del visor. La diferencia fundamental entre el buscador con respecto al visor son las siguientes:

* + El buscador no utiliza barra de menús del contenedor forma. Esto quiere decir que simplemente se omite este código al momento de implementar un componente buscador.
  + El buscador, en la barra de estado o inferior del contenedor forma, necesariamente debe tener otra barra inferior donde contener los botones de **Aceptar** y **Cancelar**. Esto se debe a que el buscador puede o no devolver a su componente primario uno o varios registros que se hayan seleccionado en el mismo.

Con estas simples modificaciones podemos tener un buscador funcional y operativo. Es evidente que su diseño y estilos son los mismos del visor.

Para la implementación del componente buscador es pertinente tener en cuenta lo siguiente:

* El componente buscador necesariamente es de tipo secundario o auxiliar. Esto quiere decir que debe ser invocado a partir de un botón. *Ver figura nº 34*.
* Pueden ser tipo lista o tipo árbol.
* Deben mostrarse en forma modal.
* Solo puede tener un botón de **Aceptar** y uno de **Cancelar**. No se debe utilizar otro tipo de verbos o sinónimos que dispersen la estandarización y la simplificación.



**Figura n° 34:** representación de la llamada a un buscador desde una ficha de datos

# Anexo I Diagrama Entidad Relación Base de Datos