

ESTÁNDARES DE PROGRAMACIÓN OFICINA VIRTUAL 2013

1. ESTÁNDARES PARA NOMBRES









- **UpperCamelCase**, cuando la primera letra de cada una de las palabras es mayúscula.
- **lowerCamelCase**, igual que la anterior con la excepción de que la primera letra es minúscula.

TIPO	NOMENCLATURA	OBSERVACIÓN
Controlador (Controller)	[objeto] Ejemplo archivo: persona, usuario, etc. Ejemplo clase: Persona, Usuario, etc.	Nombre del archivo: lowerCamelCase Nombre de la clase: UpperCamelCase
Vista (View)	[accion _view] o [panel _view] Ejemplo archivo: ins_view, upd_view, etc.	Nombre del archivo: lowerCamelCase Sólo debe existir 4 vistas: - ins_view (insertar o nuevo) - upd_view (actualizar o editar) - qry_view (listado) - panel_view (contenido centralizado)
Modelo (Model)	[objeto _model] Ejemplo archivo: persona_model, usuario_model, etc. Ejemplo clase: Persona_model, Usuario_model, etc.	Nombre del archivo: lowerCamelCase Nombre de la clase: UpperCamelCase <i>En el caso de necesitar métodos adicionales, por ejemplo: un combo para cargos, sería: cboCargosGet</i>
Métodos (Methods)	[objetoMetodo] Ejemplo método: personalns, usuario, personaGet (obtener un solo registro), personaQry (listado)	Nombre del método: lowerCamelCase <i>Los métodos de controlador y el modelo tendrán el mismo estándar.</i>
Carpetas (Folders)	[Sistema o proyecto] Ejem: sistram, sigma, etc	Solo minúsculas.
Procedimientos Almacenados (Stored Procedures)	[USP_SIGLASISTEMA_TIPOOPERACIÓN_Objeto] Ejm: USP_SIT_S_Vehiculo USP_SIT_I_Vehiculo USP_SIT_U_Vehiculo	Todo en mayúscula a excepción del Objeto que sólo la primera letra es mayúscula (Tipo de operación= U,I,S) • U (Update) • I (Insert) • S (Select, cualquier lectura de la BD)
Controles XHTML	Ejemplos formulario insertar y actualizar Nombre formulario : frm_ins_persona Caja texto para nombre : txt_ins_per_nombre Combo sexo de persona : cbo_ins_per_sexo Radio button de sexo : rdo_ins_per_sexo Check de intereses : chk_ins_per_intereses Botón registrar : btn_ins_per Dialogo de un form de edición (div que contiene al form) : c_frm_upd_persona Dialogo que contiene un combo (div que contiene el combo) : c_cbo_upd_per_sexo Ejemplos Formulario búsqueda Caja de búsqueda : txt_fnd_per_dni Botón de búsqueda : btn_fnd_per_dni Combo en el buscar : cbo_fnd_per_sexo Anchor Enlace ver datos persona: anc_fnd_per_datos	Se utilizará como ejemplo un CRUD para persona. [tipocontrol_accion_entidad_campodelform] acn = anchor (enlace) fnd = find (buscar) c = container (contenedor) <i>En el caso de nombres compuestos, por ejemplo: fecha de nacimiento, sería: txt_ins_per_fecNac (notación lowerCamelCase con 3 primeras letras por cada nombre).</i>

JavaScript	<p>[abrevObjeto] Ejem: jsPersona, jsUsuario, etc.</p> <p>Para la realización de envíos POST o GET a un controlador u otro se utilizará:</p> <pre>\$.ajax({ type: "<POST o GET>", url: "<a donde enviar>", cache: false, data: { item:\$("#item").val(), accion:"<accion>" }, success: function(data) { }, error: function() { } });</pre>	<p>Nombre del archivo: lowerCamelCase</p> <p><i>En el caso de necesitar usar otro documento javascript (en el caso de un update), el document debería ser: jsPersonaUpd</i></p>
------------	--	--

2. ESTÁNDARES PARA ÍCONOS

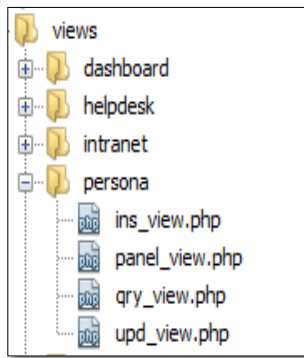
- Se debe utilizar los íconos de jQueryUI

DESCRIPCIÓN	ICONO NORMAL	ICONO HOVER
Nuevo		
Editar		
Eliminar		
Dar de Baja		



3. ESTÁNDARES PARA LOS ARCHIVOS

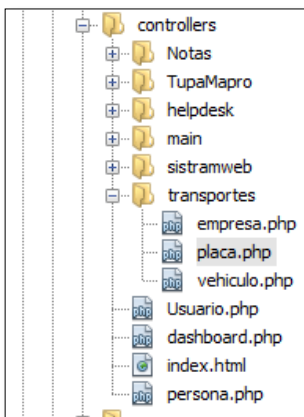
a. VISTAS (VIEWS)



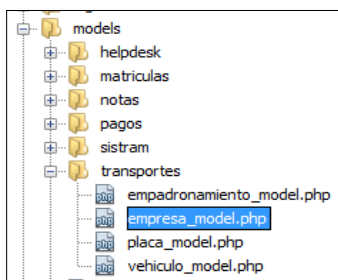
- Toda vista que pertenezca a un sistema en particular deberá ser incluida dentro de una carpeta.

- También será necesario crear carpetas para cada uno de los objetos y estas a su vez deberán tener 4 archivos básicos como se observa en la imagen de la izquierda.

b. CONTROLADORES (CONTROLLERS)

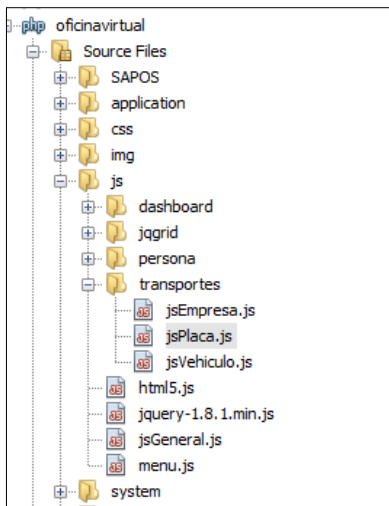


c. MODELOS (MODELS)



En la carpeta **models** las únicas carpetas que deben existir son aquellas con el nombre del sistema.

d. JAVASCRIPT (JS)



Consideraciones importantes !

- Utilizar transacciones cuando se hace varios inserts a la vez (en CI : batch).
- Evitar, en lo posible, utilizar los eventos nativos de javascript, priorizando los eventos propios de jQuery.
- Para hacer referencia a las acciones (insertar, eliminar, actualizar, buscar, etc.) se utilizará una variable llamada **\$accion**.
- Todas las imágenes que se utilicen deberán estar ubicadas dentro de la carpeta **img** que está en **application/img**; no usar subcarpetas.
- Utilizar la función **Trim()** de php para cualquier *insert* o *update*, otra opción es usar *ltrim,rtrim* desde Sql Server.
- Para recoger el resultado de una operación se utilizarán números que indiquen el estado de la misma:
 - 1 = éxito (texto a mostrar: operación realizada con éxito).
 - 0 = error (texto a mostrar: ha ocurrido un error, vuelva a intentarlo).
 - 2 = vacío (texto a mostrar: no se encontraron registros).