ESTÁNDARES DE CODIFICACIÓN PARA PLATAFORMA WEB.

El presente estandar define la utilización de la codificación web mediante el framework catalyst para mod_perl. Catalyst trabaja sobre el modelo de arquitectura de software MVC, el cual define cada una de estas siglas como sigue:

Modelo: Todo lo referente a la conexión con la base de datos.
Vista: Todo lo referente a la interfaz con el usuario.
Controlador: Realiza las acciones de control necesarias intermedias entre la vista y el modelo.

Estructura de directorios
El Departamento de Servicios Electrónicos e Internet (DSEI) proporciona la ruta sobre la cual se va a trabajar como directorio raíz del servidor web. A partir de esta, la estructura de directorios recomendada para trabajar en el catalyst es la siguiente:

1. **docs.** Directorio donde se encontrarán los archivos necesarios para el funcionamiento de la Vista. Se creará una carpeta que será el nombre del proyecto al que pertenezcan los archivos pero en minúsculas, este tendrá la siguiente estructura:
   a. **imágenes.** Directorio donde se guardarán todas las imágenes utilizadas por el sistema.
   b. **pantallas.** Directorio donde se guardarán todas las pantallas o templates utilizados por el sistema.
   c. **css.** Directorio donde se guardarán las hojas de estilos.
   d. **js.** Directorio donde se guardarán las librerías de JavaScript.

2. **NombreDelProyecto.** La estructura de directorios la define el catalyst al crear el proyecto mediante el script catalyst.pl. La raiz del mismo será el nombre del proyecto, a través del cual se accederá a internet, recordando que es sensible a mayúsculas y minúsculas y que deberá iniciar con mayúscula. De esta estructura hay que tener las siguientes consideraciones:
   a. **NombreDelProyecto/lib/NombreDelProyecto:** En esta ruta se encontrará una carpeta concernientes a cada parte del modelo MVC donde se encontraran las librerías correspondientes a las mismas.
   b. **NombreDelProyecto/root/src:** En esta ruta se encontrarán las pantallas o templates que se utilizarán para el despliegue de la vista del Proyecto.

Codificación

1. En caso de utilizar cgi's el nombre del archivo principal siempre deberá llamarse index.cgi.

2. El nombre de los archivos deberá ser en minúsculas.

3. La parte inicial de los archivos deberá llevar la siguiente información:
   - Nombre del programa
   - Nombre del desarrollador
   - Fecha de elaboración
   - Descripción
   - Administración de cambios.
     i. Autor del cambio.
     ii. Fecha.
     iii. Descripción del cambio.
     iv. Número de Solicitud.
     v. Versión que origina.
     vi. Versión del estándar aplicado.

4. En las librerías utilizadas siempre se deberá incluir la instrucción strict.

5. Variables.
   - Los nombres de las variables deberán ser escritos con la primera letra en minúscula, y las siguientes palabras que conformen el nombre, deberán llevar la primera letra en mayúscula y las demás en minúsculas a excepción de las siglas.
     i. Por ejemplo: EvaluacionDocente, CicloEscolarInicio.
   - El nombre tiene que ser claro al valor que se pretende controlar.
   - Todas las variables deberán tener un alcance definido y descrito de la siguiente manera:

COPIA NO CONTROLADA

i. Variables globales: se utilizará el prefijo g. Ejemplo: gNombreVariable.
ii. Variables Locales: se utilizará el prefijo l. Ejemplo: lNombreVariable

- Las variables utilizadas en los ciclos deberán llamarse como cualquier letra del alfabeto sin necesidad de describir el alcance.

6. Funciones y Procedimientos.
   - Deberán ser nombradas con la primera letra de cada palabra que conforme el nombre en mayúscula y las demás en minúsculas. No se deberán utilizar abreviaturas.
   - Al inicio de la función, se deberá documentar una breve descripción de lo que hace, qué datos recibe, qué resultado genera y el nombre del autor.
   - Se deberá evitar la repetición de funciones en distintos archivos, para lo cual se deberán incluir en la librería correspondiente.

7. Librerías.
   - Deberán ser nombradas con la primer letra de cada palabra en mayúscula y las demás en minúsculas obligatoriamente, además no se deberán utilizar abreviaturas.
   - La parte inicial de la librería deberá llevar la siguiente información.
     i. Nombre de la librería
     ii. Nombre del desarrollador
     iii. Fecha de elaboración
     iv. Descripción de la librería
     v. Administración de cambios

     1. Autor del cambio.
     2. Fecha.
     3. Descripción del cambio.
     4. Número de Solicitud.
     5. Versión que origina.
     6. Versión del estándar aplicado

8. Identación.
   - La identación de cada bloque (if, for, etc) o etiquetas deberá ser el valor default por el tab en el editor utilizado.

9. Acceso a Base de datos (Modelo)
   - El código de acceso a base de datos deberá estar en la carpeta NombreDelSistema/lib/NombreDelSistema/Model
   - Se creará una librería por base de datos y una función por procedimiento almacenado.
   - La variable que contiene la consulta del procedimiento almacenado se deberá llamar proc.
   - La variable que contendrá la información de respuesta a la consulta del procedimiento almacenado deberá llamarse salida.
   - La librería para acceder a la base de datos utilizada deberá ser Datos.pm (elaborada por DSEI) mediante el cual se crea el objeto de conexión en catalyst de la siguiente manera:

     ```
     my $data = Proyecto::Model::Datos->new_catalyst($self->dbh);
     ```

     donde Proyecto::Model::Datos es el namespace donde se encuentra la libreria Datos.pm. La documentación sobre su uso se encuentra en la misma librería.

   - Cada librería contendrá un paquete de configuración en el cual se especifica el servidor a conectarse, usuario y contraseña.

     ```
     __PACKAGE__->config(
       dsn          => 'dbi:Sybase:server=SERVIDOR',
      user          => 'usuario',
       password     => 'contraseña',
      options       => {},
     );
     ```

10. Características generales de HTML
    a. Las plantillas deberán tener la siguiente documentación
       i. Número de versión
       ii. Descripción de la plantilla
       iii. Nombre de quien la desarrolló
       iv. Fecha de elaboración

COPIA NO CONTROLADA

v. Versión del estándar aplicado

b. De las tablas
   i. Deberán de tener la propiedad class a la que corresponda según lo definido en la hoja de estilos proporcionada por DSEI.
   ii. En caso de necesitar bordes, la tabla deberá llevar la propiedad style con el siguiente valor: "border-collapse: collapse".

c. No se permite la utilización de frames

d. La etiqueta title, deberá de contener el nombre del sistema.

e. Deberá estar definido el archivo de hoja de estilos proporcionado por DSEI.

11. Características de los formularios de HTML

- El method, siempre deberá de ser de tipo “POST”.
- Todos los campos, sin excepción, deberán contar con las propiedades id y name, los cuales incluyen el nombre del campo, con el fin de garantizar la compatibilidad entre navegadores.
- Los campos se deberán llamar dependiendo de su tipo y con la letra inicial de cada palabla en mayúscula.

i. Tipo “Text”, txtNombreCampo.
ii. Tipo “Hidden”, hdnNombreCampo
iii. Tipo “Radio”, optNombreCampo.
iv. Tipo “Select”, cmbNombreCampo.
v. Tipo “Textarea”, areaNombreArea.
vi. Tipo “Reset”, rstNombreCampo
vii. Tipo “button”, btnNombreBoton

1. Cuando la acción del button tenga que ver con la base de datos los valores generales para el button deberán ser:
   a. Guardar. Para guardar información de la base de datos.
   b. Eliminar. Para eliminar información de la base de datos de manera lógica y física.
   c. Consultar. Para consultar información de la base de datos.
   d. Modificar. Para modificar información de la base de datos.
   e. Buscar. Para realizar búsquedas en la base de datos.

- El envio de datos (submit) deberá hacerse mediante JavaScript con la funcion submit de la forma mediante el evento onClick del button.

12. Validación.

- Se deberán validar cada uno de los campos que son indispensables que contengan datos, para evitar incompatibilidad con los tipos de datos de la tabla correspondiente de la base de datos.

COPIA NO CONTROLADA