-Módulos o controladores a agregar/modificar:

1. id=”row”
2. ~~registro (no está para el smarty y no le funca el getLibrary)~~ (Falta probar email y activación)
3. Menú
4. Perfil
5. Post y Páginas
   * Modificar tabla Post
     + post(idpost, tipo(post, page), estado, idrevision, slug, autor, orden)
     + revisión\_post(idrevision, idpost, titulo, contenido, imagen, autor, fecha, extracto, template, padre).
     + Para ambos en la carga el slug se “genera” solo en relación al título.
     + En la edición se debe cambiar manualmente.
   * Post
     + Agregar: Título, contenido, foto, categoría/s, etiqueta/s, estado(borrador, visible), visibilidad (público, privado), extracto personalizado.
     + template = post, padre = posts.
   * Páginas
     + Agregar: Título, contenido, foto, categoría/s, etiqueta/s, estado(borrador, visible), visibilidad (público, privado), extracto personalizado, template, padre.
     + En un futuro probar usar el js draggeable para manejo de plantillas.
   * Postáginas
     + recibe link y busca en la base, si no existe, al inicio.
6. Eventos (Ver lo que se hizo con NV)
7. Configuración
   * ~~Crear módulo “configuracion”.~~
   * ~~Poner excepción en controlador “Módulos”.~~
   * Datos del sitio.
     + Nombre
     + App Slogan
     + App Company
   * ~~Usuarios~~
     + ~~Pasar el módulo a Controlador único dentro de la carpeta~~
     + ~~Separar “login” y “registro”.~~
     + Campos dinámicos para usuarios.
   * ~~Acl~~
   * Páginas (inicio(controlador, módulo o página), posts)
     + Crear tabla en la BD
   * Themes
     + Functions
     + Icon
     + Favicon
     + Lector XML para Themes
   * Menúes
     + Trabajar con el Widget o una sección específica para los menúes.
   * ~~Módulos~~
     + ~~Lectura de directorio para Modulos.~~
     + ~~búsqueda de la BD si están instalados o no.~~
       - ~~tablas:~~
         * ~~modulos(idmodulo, nombre, descripción,autor,versión, habilitado)  
           CK (nombre).~~
     + ~~Controlador que muestre la lista de módulos, estén instalados o no y que permita editar eso.~~
     + ~~Modificar Request para que permita o no usar el módulo.~~
     + ~~Configurar Accesos a usuarios.~~
   * ~~Widgets~~
     + ~~Instaladores de Widgets, que lea el directorio y decida si los va usar o no.~~
       - ~~Lectura de directorio para widgets.~~
       - ~~búsqueda de la BD si están instalados o no.~~
         * ~~tablas:~~

~~widgets(idmodulo, nombre, descripción,autor,versión, habilitado)~~

~~CK (nombre).~~

* + - * ~~Controlador que muestre la lista de widgets, estén instalados o no y que permita editar eso.~~
      * ~~Modificar views para que permita o no usar el módulo.~~
      * ~~Configurar Accesos a usuarios.~~
      * ~~Lector json para datos del Widget~~
      * ~~Leer desde la db en View.php~~
  + Controladores en gral.

1. Gestión de archivos
2. Formularios de Contacto
3. Multidioma?
4. Themes.
5. PopUps?

Modificaciones del Framework

1. **Importantes:**
   * Seguridad
   * ORM (http://propelorm.org/)
   * i18n (Intl)
2. ~~Instalar TinyMCE.~~(Solo en default\_theme).
3. ~~Instalar Foundation.~~
4. ~~Lectura de directorio para módulos.~~
5. ~~Cambiar themes por templates.~~
   * ~~Cambiar layout por “themes”.~~
     + ~~crear carpeta de theme por defecto.~~
     + ~~Forma del theme~~
       - ~~css~~
       - ~~img~~
       - ~~js~~
       - ~~configs~~
       - ~~default.tpl~~
       - ~~home.tpl~~
       - ~~configs.php~~
   * ~~Cambios en View.php~~
     + ~~Agregar attr $\_theme y setear por defecto~~
     + ~~En función render~~
       - ~~editar el template\_dir~~
       - ~~editar el config\_dir~~
       - ~~editar el $\_params~~
     + ~~editar el getLayoutPositions~~
       - ~~editar el display del render~~
     + Hacer el setTheme y editar el setTemplate
6. Drag&Drop Upload.
7. ~~Separar Widgets por carpeta.~~
   * ~~Meter menú en carpeta.~~
   * ~~Editar widget y widgetMethod~~
   * ~~editar loadModel y render~~
   * ~~Cambiar documentación.~~

# Cambios concretos

Modificaciones en MySQL

1. ~~Renombrar “modulos” por “modules”.~~
2. ~~Renombrar “permisos” por “permissions”.~~
3. ~~Renombrar “permisos\_role” por “permissions\_role”.~~
4. ~~Renombrar “permisos\_usuario” por “permissions\_user”.~~
5. ~~Renombrar “usuarios” por “users”.~~
6. En tabla “menues”.
   * ~~“nombre” por “name”~~
   * ~~“habilitado” por “enable”.~~
7. En tabla “menu\_items”.
   * ~~“idpadre” por “parent”.~~
   * ~~“orden” por “ítem\_order”.~~
8. En table “modules”.
   * “~~idmodulo” por “idmodule”~~
   * “~~carpeta” por “folder”~~
   * ~~“nombre” por “name”~~
   * ~~“descripción” por “description”~~
   * ~~“autor” por “author”~~
   * ~~“habilitado” por “enable”~~
9. En table “permissions”.
   * ~~“idpermiso” por “idpermission”~~
   * ~~“permiso” por “name”~~
   * ~~“llave” por “permission\_key”~~
10. En table “permissions\_role”.
    * ~~“idpermisorole” por “idpermissionrole”~~
    * ~~“idpermiso” por “idpermission”~~
    * ~~“valor” por “value”~~
11. En table “permissions\_user”.
    * ~~“idpermisousuario” por “idpermissionuser”~~
    * ~~“idsuario” por “iduser”~~
    * ~~“idpermiso” por “idpermission”~~
    * ~~“valor” por “value”~~
12. En table “users”.
    * ~~“idusuario” por “iduser”~~
    * ~~“nombre” por “name”~~
    * ~~“usuario” por “user”~~
    * ~~“estado” por “enable”~~
    * ~~“fecha” por “date”~~
    * ~~“código” por “code”~~
13. En table “widgets”.
    * ~~“carpeta” por “folder”~~
    * ~~“nombre” por “name”~~
    * ~~“descripción” por “description”~~
    * ~~“autor” por “author”~~
    * ~~“habilitado” por “enable”~~
14. En table “tags”
    * “name” por “tag”.
15. Deletes
    * ~~Table “categories”~~
      + ~~Tabla completa~~
    * ~~En table “permissions\_role”.~~
      + ~~idpermissionrole~~
    * ~~En table “permissions\_user”.~~
      + ~~idpermissionuser~~
    * ~~En table “posts\_categories”.~~
      + ~~idpost\_category~~
    * ~~En table “tags”~~
      + ~~title~~
16. Constraints
    * ~~En table “comments”.~~
      + ~~idpost FK reference posts(idpost)~~
    * En table “menú\_items”
      + ~~idmenu FK reference menú(idmenu)~~
      + ~~parent FK reference menu\_item(idmenuitem)~~
    * En table “permissions\_role”
      + ~~idrole FK reference role(idrole)~~
      + ~~idpermission FK reference permissions(idpermission)~~
      + ~~PK(idrole,idpermission)~~
    * En table “permissions\_users”
      + ~~iduser FK reference users(iduser)~~
      + ~~idpermission FK reference permissions(idpermission)~~
      + ~~PK(idrole,iduser)~~
    * En table “posts”
      + ~~parent FK reference posts(idpost)~~
      + ~~autor FK reference users(iduser)~~
    * En table “posts\_categories”
      + ~~idpost FK reference posts (idpost)~~
      + ~~idcategory FK reference category(idcategory)~~
      + ~~PK(idpost,idcategory)~~
    * En table “posts\_tags”
      + ~~idpost FK reference posts (idpost)~~
      + ~~idtag FK reference tags(idtag)~~
      + ~~PK(idpost,idtag).~~
    * En table “posts\_revisions”
      + ~~idpost FK reference posts (idpost)~~
      + ~~autor FK reference users(iduser)~~
    * En table “show\_menues”
      + ~~idmenu FK reference menues(idmenu)~~
    * En table “users”
      + ~~idrole FK reference roles(idrole)~~
    * En table “widgets\_content”
      + ~~idwidget FK reference widgets(idwidet)~~
    * En table “categories”
      + ~~parent FK reference categories(idcategory)~~

# Cambios concretos para seguridad

* Registrar transacciones con fecha/hora
* Usar Log de Propel
* ~~Crear usuario en BD con menos privilegios~~
  + Asignar Usuario al Framework
* Open basedir para restringir accesos a archivos
* Permisos limitados al usuario web de PHP.
* Establecer conexiones sobre SSL para encriptar las comunicaciones cliente/servidor
* Establecer conexiones sobre SSH para encriptar la conexión de red entre los clientes y el servidor de bases de datos.
* Usar SHA-2.
* Revisar las consultas para evitar inyecciones SQL.
  + Nunca se conecte como superusuario o como propietario de la base de datos. Siempre utilice usuarios personalizados con privilegios muy limitados.
  + Emplee sentencias preparadas con variables vinculadas. Son proporcionadas por PDO, MySQLi y otras bibliotecas.
  + Compruebe si la entrada proporcionada tiene el tipo de datos previsto. PHP tiene un amplio rango de funciones para validar la entrada de datos, desde las más simples, encontradas en Funciones de variables y en Funciones del tipo carácter (p.ej., is\_numeric(), ctype\_digit() respectivamente), hasta el soporte para Expresiones regulares compatibles con Perl.
  + Si la expresión espera una entrada numérica, considere verificar los datos con la función ctype\_digit(), o silenciosamente cambie su tipo utilizando settype(), o emplee su representación numérica por medio de sprintf().
* Archivo de notificación de errores
  + Una forma de detectar este problema por adelantado es hacer uso de la función propia de PHP error\_reporting()